

AUTOMATIZAÇÃO DO DIMENSIONAMENTO GRAVÍTICO E SÍSMICO DE ESTRUTURAS METÁLICAS

JOÃO PEDRO RODRIGUES NOGUEIRA

Dissertação submetida para satisfação parcial dos requisitos do grau de
MESTRE EM ENGENHARIA CIVIL — ESPECIALIZAÇÃO EM ESTRUTURAS

Orientador: Professor Doutor José Miguel de Freitas Castro

Coorientador: Engenheiro Luís Augusto Ferreira Rodrigues de
Macedo

JULHO DE 2015

MESTRADO INTEGRADO EM ENGENHARIA CIVIL 2014/2015

DEPARTAMENTO DE ENGENHARIA CIVIL

Tel. +351-22-508 1901

Fax +351-22-508 1446

✉ miec@fe.up.pt

Editado por

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Rua Dr. Roberto Frias

4200-465 PORTO

Portugal

Tel. +351-22-508 1400

Fax +351-22-508 1440

✉ feup@fe.up.pt

🌐 <http://www.fe.up.pt>

Reproduções parciais deste documento serão autorizadas na condição que seja mencionado o Autor e feita referência a *Mestrado Integrado em Engenharia Civil - 2014/2015 - Departamento de Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2015.*

As opiniões e informações incluídas neste documento representam unicamente o ponto de vista do respetivo Autor, não podendo o Editor aceitar qualquer responsabilidade legal ou outra em relação a erros ou omissões que possam existir.

Este documento foi produzido a partir de versão eletrónica fornecida pelo respetivo Autor.

Aos meus Pais e irmã

*“Perfection is achieved, not when there is nothing more to add, but when there is nothing left
to take away.”*

Antoine de Saint-Exupéry

AGRADECIMENTOS

Ao Professor José Miguel Castro pela confiança que sempre depositou em mim, por me ter orientado num trabalho pouco convencional e ter demonstrado sempre disponibilidade para discutir ideias. Pela motivação ao longo do semestre e pela revisão cuidada do trabalho.

Ao Eng. Luís Macedo por ter aceitado juntar-se a este projeto e pelas horas despendidas. Em especial, pela partilha de conhecimento sobre algoritmos de otimização, sem a qual teria sido impossível concluir o trabalho dentro do prazo.

Ao António Silva pela contribuição que deu na conceptualização do trabalho e por me ter cedido o *template* da sua tese.

Aos meus pais e à minha irmã que sempre me apoiaram ao longo do meu percurso e me deram todas as condições necessárias para o concluir com sucesso.

À Filipa pelo apoio e motivação que sempre me deu (e por me aturar quando as análises não corriam bem).

A todos os meus amigos pelos bons momentos que passamos juntos.

De uma forma particular aos que me acompanharam sempre de perto estes 5 anos: Zé Calejo, Zé Pedro, Mariana Moreira, Rita Martins, Joana Ferreira, Gonçalo Tiago e Diana Abrunhosa.

De uma forma geral, aos habitantes do Lote 18, meus companheiros de estudo e de aventuras; aos membros do NTIEC que abraçaram um projeto inexistente com uma equipa de estranhos e o tornaram no que hoje é; e aos colegas que me acompanharam no quinto ano, com quem partilhei bons momentos.

Por fim, aos amigos que conheço desde que me conheço: a Ana Reis e o Pedro Teixeira.

RESUMO

Este trabalho procurou o desenvolvimento de uma ferramenta informática que permitisse o dimensionamento otimizado de pórticos metálicos sob ações gravíticas e sísmicas. Dado um conjunto de dados (geometria, ações, zona sísmica, grupos de elementos, etc.) é executado um algoritmo de otimização que visa devolver uma estrutura altamente eficiente, que cumpra todos os requerimentos impostos pelo Eurocódigo 3 e pelo Eurocódigo 8 mas com o menor peso possível. Foi implementado, no algoritmo de otimização, a metodologia *Improved Force-Based Design* (IFBD) proposta por Castro, *et al.* [1], que procura ajustar o coeficiente de comportamento à estrutura, resultando assim em estruturas mais leves e económicas. Deste trabalho resultou o presente documento escrito e uma ferramenta informática.

No desenvolvimento, procurou-se recorrer apenas a *softwares Open Source* com o objetivo de manter a ferramenta livre de custos, pelo que esta foi desenvolvida em Python, apoiando-se no OpenSees para a realização das análises estruturais. Foi, também, tido o cuidado de a estruturar de forma modular para que a inclusão de novas funcionalidades pudesse ser relativamente fácil para outros utilizadores.

Com o objetivo de aproveitar as capacidades do OpenSees e estudar o comportamento das estruturas de forma mais realista, todas as análises realizadas são de segunda ordem, considerando os efeitos $P-\Delta$ na estrutura.

Por fim, para testar a qualidade dos resultados, são realizados vários casos de estudo nos quais se dimensiona com a ferramenta alguns pórticos dimensionados manualmente e publicados em trabalhos anteriores. Estes resultados são, posteriormente, comparados com os provenientes dos dimensionamentos manuais.

Palavras-Chave: Pórticos Metálicos, Python, OpenSees, Eurocódigo 3, Eurocódigo 8, Improved Force-Based Design, Algoritmos de Otimização, Pesquisa Harmónica, Análise não-linear, efeitos $P-\Delta$

ABSTRACT

This thesis aimed at the development of a tool capable of optimizing the design of steel frames under gravity and seismic loads. For a given input (geometry, loads, seismic information, groups of elements, etc.) an optimization algorithm is executed in order to obtain a highly efficient structures, which meets all the requirements imposed by EC3 and EC8 with the lowest weight possible. The Improved Force-Based Design (IFBD), published by Castro, et al. [1], was also implemented. As a result, this thesis is composed of both this document and the tool itself.

An effort was made in order to use only Open Source resources, so as to create a tool that is completely free. The code was developed in Python, using OpenSees as the structural analysis engine. It was also taken into account the need to develop a modular structure to be easy to implement new functionalities.

As OpenSees is very reliable for non-linear analysis, every analysis made takes into consideration P- Δ effects on the structure. Which allows for a more realistic simulation of behavior and, therefore, to a better and lighter structure.

To demonstrate the capabilities of the tool created, a series of tests were made on some structures already designed on other papers and the results were compared and presented.

KEYWORDS: Steel Frames, Python, OpenSees, Eurocode 3, Eurocode 8, Improved Force-Based Design, Evolutionary Algorithms, harmonic Search, Non-linear analysis , P- Δ effects.

ÍNDICE GERAL

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
 1. INTRODUÇÃO	 1
1.1. ENQUADRAMENTO GERAL.....	1
1.2. OBJETIVOS	2
1.3. ESTRUTURA DA DISSERTAÇÃO	2
 2. DIMENSIONAMENTO OTIMIZADO DE ESTRUTURAS METÁLICAS	 5
2.1. INTRODUÇÃO	5
2.2. EUROCÓDIGO 3.....	5
2.2.1. GENERALIDADES	5
2.2.2. PROPRIEDADES DOS MATERIAIS.....	5
2.2.3. ANÁLISE GLOBAL.....	6
2.2.4. CLASSIFICAÇÃO DAS SEÇÕES TRANSVERSAIS.....	7
2.2.5. ESTADOS LIMITES DE UTILIZAÇÃO.....	7
2.2.6. ESTADO LIMITE ÚLTIMO	8
2.3. EUROCÓDIGO 8.....	10
2.3.1. GENERALIDADES	10
2.3.2. REQUISITOS DE DESEMPENHO E CRITÉRIOS DE CONFORMIDADE	10
2.3.3. CARACTERIZAÇÃO DA AÇÃO SÍSMICA	10
2.3.4. MÉTODO DE ANÁLISE POR FORÇAS LATERAIS EQUIVALENTES	12
2.3.5. VERIFICAÇÕES DE SEGURANÇA	12
2.3.5.1. Estado limite último	12
2.3.5.2. Sensibilidade do deslocamento relativo entre pisos	13
2.3.5.3. Condição de ductilidade local	14
2.3.5.4. Limitação de danos	14
2.3.6. COEFICIENTE DE COMPORTAMENTO - IMPROVED FORCE-BASED DESIGN	15
2.4. ALGORITMOS DE OTIMIZAÇÃO	16
2.4.1. PESQUISA HARMÓNICA.....	16

2.5. CONSIDERAÇÕES FINAIS	19
--	-----------

3. DESENVOLVIMENTO DE UMA FERRAMENTA PARA CÁLCULO OTIMIZADO DE ESTRUTURAS METÁLICAS 21

3.1. DEFINIÇÃO DOS OBJETIVOS	21
---	-----------

3.2. MODELO DE INFORMAÇÃO	21
--	-----------

3.2.1. INTRODUÇÃO.....	21
------------------------	----

3.2.2. INTRODUÇÃO DE DADOS	23
----------------------------------	----

3.2.2.1. Geometria e Propriedades Gerais	23
--	----

3.2.2.2. Ações.....	24
---------------------	----

3.2.2.3. Ação Sísmica	24
-----------------------------	----

3.2.2.4. Perfis a utilizar.....	25
---------------------------------	----

3.2.2.5. Grupos de pilares/vigas.....	25
---------------------------------------	----

3.2.3. GERAÇÃO DO MODELO.....	25
-------------------------------	----

3.2.3.1. Pisos recuados.....	26
------------------------------	----

3.3. ANÁLISES ESTRUTURAIS	26
--	-----------

3.3.1. OPENSEES	26
-----------------------	----

3.3.2. CRIAÇÃO DOS MODELOS DE ANÁLISE	27
---	----

3.3.3. ANÁLISE DINÂMICA	28
-------------------------------	----

3.3.4. COMBINAÇÕES	28
--------------------------	----

3.3.5. PROCESSAMENTO E ARMAZENAMENTO DOS RESULTADOS	29
---	----

3.4. VERIFICAÇÕES DE SEGURANÇA.....	29
--	-----------

3.4.1. ESTADO LIMITE DE SERVIÇO	29
---------------------------------------	----

3.4.2. ESTADO LIMITE ÚLTIMO.....	30
----------------------------------	----

3.4.2.1. Verificações seccionais	30
--	----

3.4.2.2. Verificações à encurvadura	30
---	----

3.4.3. DESENVOLVIMENTO DO MÓDULO DE VERIFICAÇÃO DE SEGURANÇA	30
--	----

3.4.3.1. Secções metálicas	31
----------------------------------	----

3.4.3.2. Elementos metálicos	32
------------------------------------	----

3.5. ALGORITMOS DE OTIMIZAÇÃO	34
--	-----------

3.5.1. ALGORITMO BÁSICO	34
-------------------------------	----

3.5.2. PESQUISA HARMÓNICA	35
---------------------------------	----

3.5.2.1. Formulação do problema.....	35
--------------------------------------	----

3.5.2.2. OBTENÇÃO DOS PARÂMETROS DE CARACTERIZAÇÃO DA PESQUISA HARMÓNICA	37
--	----

4. DOCUMENTAÇÃO DA FERRAMENTA DE CÁLCULO	41
4.1. INTRODUÇÃO	41
4.2. CLASSE MODEL	41
4.2.1. MÉTODO __INIT__	42
4.2.2. MÉTODO _DESIGN_CONDITION	43
4.2.3. MÉTODO _REDUCE_IN_HEIGHT	43
4.2.4. MÉTODO TOTAL_WEIGHT	43
4.2.5. MÉTODO BASIC_DESIGN	43
4.3. CLASSE INTERFACE	44
4.3.1. LEITURA DE DADOS E CRIAÇÃO DO MODELO	44
4.3.2. EXPORTAÇÃO DE RESULTADOS	45
4.3.3. INTERFACE COM O OPENSEES	46
4.4. CLASSE GRAVITIC	47
4.4.1. MÉTODO GRAVITIC_VERIFICATION	47
4.4.2. MÉTODO GRAVITIC_ANALYSIS_AND_OPTIMIZATION	47
4.5. CLASSE DYNAMIC	47
4.5.1. INTRODUÇÃO DE DADOS E OBTENÇÃO DAS PROPRIEDADES SÍSMICAS	47
4.5.2. MÉTODO SEISMIC_ANALYSIS_OS	48
4.5.3. VERIFICAÇÕES DE SEGURANÇA	48
4.5.3.1. Método seismic_check_ULS	48
4.5.3.2. Método capacity_design_check	49
4.5.3.3. Método seismic_check_SLS	49
4.5.3.4. Método interstorey_drift_sensitivity	49
4.5.4. MÉTODO SEISMIC_VERIFICATION	50
4.5.5. MÉTODO SEISMIC_ANALYSIS_AND_OPTIMIZATION	50
4.6. CLASSE OPTIMIZATION	51
4.6.1. MÉTODOS DA CLASSE OPTIMIZATION	51
4.6.2. MÉTODO HS_OBJECTIVEFUNCTION	51
4.7. OUTRAS CLASSES	52
4.7.1. CLASSE NODE	52
4.7.2. CLASSE _ELEMENT	52
4.7.2.1. Verificações gravíticas	53
4.7.2.2. Verificações sísmicas	53
4.7.3. CLASSE LOAD	54
4.7.4. CLASSE GROUP	54

4.8. MÓDULO DE VERIFICAÇÕES DE SEGURANÇA	54
4.8.1. CLASSE SECTION	55
4.8.1.1. Classe Steel_Section	55
4.8.1.2. Classe I_commercial	55
4.8.2. CLASSE ELEMENT.....	55
4.8.2.1. Classe Steel_Element	56
 5. CASOS DE ESTUDO	 59
5.1. INTRODUÇÃO	59
5.2. CASOS DE ESTUDO	59
5.2.1. PÓRTICOS DE 3 PISOS	61
5.2.2. PÓRTICOS DE 5 PISOS	62
5.2.3. PÓRTICOS DE 8 PISOS	63
5.3. COMPARAÇÃO DOS PÓRTICOS ESTUDADOS.....	66
 6. CONSIDERAÇÕES FINAIS	 67
6.1. SUMÁRIO E CONCLUSÕES.....	67
6.2. DESENVOLVIMENTOS FUTUROS.....	67
 REFERÊNCIAS BIBLIOGRÁFICAS	 69

ÍNDICE DE FIGURAS

Fig. 2.1 - Deslocamentos verticais em vigas [6]	7
Fig. 2.2 - Espectro de resposta elástico [2].....	11
Fig. 2.3 - Curva de resposta lateral [1].....	16
Fig. 2.4 - Analogia entre uma banda de Jazz e um problema de dimensionamento	17
Fig. 2.5 - Esquema de funcionamento da pesquisa harmónica	19
Fig. 3.1 - Esquema global de funcionamento do programa	22
Fig. 3.2 - <i>Screenshot</i> da folha de Excel para introdução dos dados gerais.....	23
Fig. 3.3 - Excerto da folha de Excel utilizada para introdução de ações nos elementos.....	24
Fig. 3.4 - Excerto da folha de Excel utilizada para introdução de ações sísmicas.....	24
Fig. 3.5 - Excerto da folha de Excel utilizada para a definição dos grupos de pilares	25
Fig. 3.6 - Exemplificação do sistema de catalogação.....	26
Fig. 3.7 - Estrutura modular do OpenSees [15].....	27
Fig. 3.8 - Esquema da arquitetura idealizada para o módulo.....	31
Fig. 3.9 - Exemplo de aplicação da classificação a uma viga IPE600 com $N_{ed} = -1000\text{kN}$ e $M_{ed} = 300\text{kNm}$	32
Fig. 3.10 - Regressão linear de C_1	33
Fig. 3.11 - Fluxograma do processo associado ao método básico	34
Fig. 3.12 - Estrutura de 3 vão com 5 pisos.....	36
Fig. 3.13 - Evolução dos parâmetros proposta por Kumar, <i>et al.</i> [11]	37
Fig. 3.14 – Representação gráfica dos parâmetros utilizados no presente trabalho	39
Fig. 4.1 - Esquema global de funcionamento do programa	42
Fig. 4.2 - Representação da estrutura introduzida pelo utilizador	45
Fig. 4.3 – Exemplo de <i>Output</i> gráfico de uma pesquisa harmónica	45
Fig. 4.4 - Fluxograma do método seismic_verification.....	50
Fig. 4.5 - Diagrama de classes do módulo de verificação	56
Fig. 5.1 - Representação do edifício cujo pórtico pertence, para o caso de 5 pisos. [23]	60
Fig. 5.2 - Distribuição das ações gravíticas num piso do pórtico [24].....	60
Fig. 5.3 - Definição dos grupos para o pórtico de 3 pisos.....	61
Fig. 5.4 - Definição dos grupos para o pórtico de 5 pisos.....	62
Fig. 5.5 - Definição dos grupos para o pórtico de 5 pisos.....	64
Fig. 5.6 - Gráfico comparativo dos pesos das soluções obtidas	66

ÍNDICE DE TABELAS

Tabela 1 - Valores nominais de tensão de cedência f_y e da tensão última f_u para aços estruturais laminados a quente (excerto) [4]	6
Tabela 2 - Analogia dos parâmetros de caracterização da pesquisa harmónica.....	17
Tabela 3 - Combinações de dimensionamento geradas pelo programa	28
Tabela 4 - Excerto da Tabela 6.11 do <i>Designer's guide to EC3</i> [17].....	33
Tabela 5 - Cálculo dos critérios de penalização	36
Tabela 6 - Resultados dos testes de sensibilidade dos parâmetros do HS.....	38
Tabela 7 - Ações gravíticas para o dimensionamento do pórtico.....	60
Tabela 8 - Comparação dos resultados obtidos entre o dimensionamento manual e o otimizado para os pórticos de 3 pisos	61
Tabela 9 - Comparação dos resultados obtidos entre o dimensionamento manual e o otimizado para os pórticos de 5 pisos	63
Tabela 10 - Comparação entre os dimensionamentos automático e manual utilizando a metodologia IFBD	65
Tabela 11 - Comparação entre os dimensionamentos automático e manual para um pórtico de classe DCM	65
Tabela 12 - Comparação entre os dimensionamentos automático e manual para um pórtico de classe DCH.....	66

SÍMBOLOS, ACRÓNIMOS E ABREVIATURAS

Letras Latinas Maiúsculas

A – área de uma secção transversal

E - módulo de elasticidade [GPa]

E – ação accidental sísmica

G – módulo de distorção [GPa]

G – ação permanente

K – Kelvin

L – comprimento do elemento

Q_s – Ação variável sobrecarga

Q_v – Ação variável vento

Letras Latinas Minúsculas

b – largura de uma secção transversal

f_y – tensão de cedência

h_w – altura da alma

q – coeficiente de comportamento

t_w – espessura da alma

t_f – espessura do banzo

Letras Gregas

γ_G – coeficiente parcial relativo a ações permanentes

γ_M – coeficiente parcial de segurança

γ_{ov} – coeficiente de sobrerresistencia do material

γ_Q – coeficiente parcial relativo a ações variáveis

θ – coeficiente de sensibilidade ao deslocamento relativo entre pisos

σ – tensão

ν – coeficiente de Poisson em regime elástico

ν – relação sísmico SLS com ULS

ψ₂ – coeficiente para a determinação do valor de combinação de uma ação variável

Ω – coeficiente multiplicador do esforço normal resultante da análise e devido à ação sísmica de calculo, para o cálculo dos elementos não dissipativos em pórticos

Abreviaturas

BW – bandwidth

DCM – classe de ductilidade média

DCH – classe de ductilidade alta

eBC - elasticBeamColumn

fBC - forceBeamColumn

HMCR - harmony memory considering rate

HMS – harmony memory size

HS – harmony search

OS - OpenSees

PAR – pitch adjustment rate

RC – reinforced concrete

RSA - Robot Structural Analysis

1

INTRODUÇÃO

1.1. ENQUADRAMENTO GERAL

A engenharia civil é uma atividade cuja responsabilidade social é das mais elevadas, uma vez que erros na tomada de decisões de um engenheiro civil poderão resultar na perda de milhares de euros, ou até de vidas humanas. Assim, é fundamental que as suas decisões sejam apoiadas num conhecimento o mais desenvolvido possível, daí esta atividade ser altamente regulamentada a nível mundial e os regulamentos estarem em constante atualização. Contudo, e uma vez que a maioria dos profissionais se dedica a atividades de projeto, a engenharia civil carece de *softwares* que permitam a versatilidade e liberdade de experimentação que a investigação em estrutura de aço requer. Esta versatilidade, se aliada às especificações inerentes à engenharia sísmica, torna o universo de *softwares* disponíveis no mercado praticamente nulo, dificultando o estudo de modelos que não se enquadrem em nenhum regulamento.

O aço é um material utilizado globalmente, associado à arquitetura moderna, resultado de um conjunto de fatores que tornam a sua utilização apetecível. A sua crescente popularidade deve-se, principalmente, a elevadas capacidades de resistência e ductilidade, ao seu custo e ao facto de ser um material altamente reciclável. Para além disso, muitos dos componentes de uma estrutura de aço podem ser pré-fabricados fora do local de construção, garantindo padrões de qualidade superiores a outros materiais de construção. As modificações aos componentes é relativamente simples, as propriedades dos materiais não se alteram significativamente com o tempo e o peso dos elementos é reduzido, permitindo maiores vãos.

O projeto sísmico de estruturas em Portugal encontra-se atualmente numa fase de progressiva implementação do Eurocódigo 8, uma norma claramente mais extensa e minuciosa em comparação com o enquadramento regulamentar ainda em vigor no território português. Este facto implicará mudanças ao nível do projeto, já que irá existir uma norma de aplicação nacional que contém prescrições específicas para estruturas metálicas. Para além disso, estudos recentes têm vindo a demonstrar que diversas opções passíveis de serem tomadas na aplicação do Eurocódigo 8 [2] podem ter consequências significativas na quantidade final de aço empregue na estrutura. Neste contexto, torna-se clara a necessidade de criação de ferramentas de cálculo de apoio ao projeto sísmico que permitam o dimensionamento otimizado de estruturas, assegurando ao mesmo tempo o cumprimento das várias disposições regulamentares.

O processo de dimensionamento é baseado em vários fatores como: a experiência do projetista, ensaios laboratoriais e resultados de trabalhos de investigação. Este processo tem maioritariamente que conciliar dois objetivos antagónicos: economia e segurança. Assim, o projetista tem que lidar com várias dificuldades que poderão dificultar a procura de soluções económicas, como a complexidade da estrutura

ou o uso que esta terá no futuro, levantando assim questões sobre a pertinência da criação de ferramentas de otimização do dimensionamento estrutural.

Neste trabalho, descreve-se a concepção e a implementação de um programa informático que permite o dimensionamento de pórticos metálicos sob ações gravíticas e sísmicas. O programa, integralmente desenvolvido em linguagem Python num paradigma orientado a objetos, processa um conjunto de dados de entrada e faz uso do programa de análise estrutural OpenSees [3] para a obtenção de esforços e deslocamentos da estrutura resultantes das diferentes combinações de ações. De seguida, o programa recorre a diversos algoritmos de verificação de segurança desenvolvidos pelos autores para verificar todas as prescrições regulamentares previstas nos Eurocódigos 3 [4] e 8 relativas aos vários estados limite. Com vista à obtenção de soluções eficientes do ponto de vista económico, foi implementado um algoritmo de otimização meta-heurístico. O trabalho termina com a aplicação do programa desenvolvido a um conjunto de pórticos metálicos simples, disponíveis em trabalhos científicos publicados recentemente. Os resultados obtidos permitem confirmar a eficácia do programa em obter soluções otimizadas, envolvendo poupanças de aço muito significativas em relação a dimensionamentos realizados por processos convencionais.

1.2. OBJETIVOS

A procura de soluções estruturais mais sustentáveis passa pela experimentação de milhares de modelos para que os dados possam ser correlacionados de forma expressiva. Infelizmente, a carência de *software* adequado a esta experimentação torna o processo bastante moroso e, por vezes, injustificável. Para tentar inverter esta tendência, este trabalho procura iniciar o desenvolvimento de uma ferramenta livre que permita o dimensionamento otimizado de pórticos metálicos sujeitos a cargas gravíticas e sísmicas.

Pretende-se que o desenvolvimento desta ferramenta possa ser continuado no futuro e, eventualmente, expandido a outros tipo de estruturas, sejam estas realizadas com outros materiais ou com outros tipos de comportamento estrutural. Para que a linguagem não seja um obstáculo, esta ferramenta foi desenvolvida numa linguagem de alto nível, Python, e utiliza o OpenSees para executar as análises estruturais, pois é um *software Open Source* com um elevado nível de fiabilidade. O apoio no OpenSees permitiu um estudo das estruturas considerando os efeitos $P-\Delta$, o que permite não só um estudo de comportamento mais realista como o contorno de algumas restrições dos Eurocódigos normalmente associadas às análises lineares.

1.3. ESTRUTURA DA DISSERTAÇÃO

Este trabalho é constituído por seis capítulos, cujo objetivo é enquadrar o leitor no processo de desenvolvimento da ferramenta de cálculo, e pela própria ferramenta. Ao desenvolver este documento escrito procurou-se não só informar sobre as capacidades da ferramenta, os seus pressupostos e as decisões tomadas, mas também fornecer informação sobre a arquitetura do programa, possibilitando o desenvolvimento ou modificação da ferramenta.

No segundo capítulo, **Dimensionamento otimizado de estruturas metálicas** são apresentadas as regras, prescrições e metodologias de cálculo que regem o atual dimensionamento de pórticos metálicos a ações gravíticas e sísmicas, sendo feita referência ao processo associado à implementação do método de dimensionamento sísmico *Improved Force-Based Design* (IFBD) [1]. É, também, introduzido o conceito de algoritmo de otimização e explicado o funcionamento de um algoritmo de pesquisa harmónica.

No capítulo 3, **Desenvolvimento de uma ferramenta para cálculo otimizado de estruturas metálicas** é explicada a conceptualização da ferramenta, a constituição do modelo de informação e as decisões tomadas ao longo do processo de conceção da ferramenta.

No capítulo 4, **Documentação da ferramenta de cálculo** é explicada a estrutura do programa em detalhe. Este capítulo foi escrito com uma linguagem mais técnica, por vezes associada a funções específicas do Python, pois visa integrar o leitor na estrutura do mesmo, permitindo-lhe perceber rapidamente o seu funcionamento.

No capítulo 5, **Casos de estudo** são apresentados resultados de estruturas resolvidas com recurso à ferramenta desenvolvida e comparação entre estes e estruturas previamente dimensionadas manualmente, pelo processo convencional.

No capítulo 6, **Considerações finais** são apresentadas as conclusões finais deste trabalho e as propostas do autor para desenvolvimentos futuros.

2

DIMENSIONAMENTO OTIMIZADO DE ESTRUTURAS METÁLICAS

2.1. INTRODUÇÃO

Neste capítulo apresentam-se as regras e prescrições que um projetista deverá seguir no dimensionamento de pórticos metálicos para ações base gravíticas e sísmicas, segundo as normas europeias. Assim, será introduzido o Eurocódigo 3 [4], a norma que contém as regras a seguir no dimensionamento de estruturas metálicas, e o Eurocódigo 8 [2], onde constam as regras de dimensionamento europeias para edifícios sujeitos a ações sísmicas. Será também apresentada uma introdução a algoritmos de otimização e a sua aplicação a problemas de engenharia civil.

2.2. EUROCÓDIGO 3

2.2.1. GENERALIDADES

A regulamentação europeia de estruturas metálicas, Eurocódigo 3 (EC3) inclui, na Parte 1, as regras para a análise e dimensionamento a aplicar a estruturas metálicas correntes. O EC3 pressupõe a conformidade do fabrico e montagem com a norma EN 1090 [5].

2.2.2. PROPRIEDADES DOS MATERIAIS

Os valores de cálculo das propriedades dos materiais são obtidos dividindo os valores característicos por coeficientes de segurança que, segundo o anexo nacional português, tomam os seguintes valores: $\gamma_{M0} = 1.00$, $\gamma_{M1} = 1.00$ e $\gamma_{M2} = 1.25$. A secção 3 do EC3 define os requisitos que o aço deve cumprir. Uma vez que esse estudo excede o âmbito do presente trabalho, são apresentados na Tabela 1 os valores de referência mais comuns.

Tabela 1 - Valores nominais de tensão de cedência f_y e da tensão última f_u para aços estruturais laminados a quente (excerto) [4]

Norma e classe de aço	Espessura nominal t do componente da secção [mm]			
	$t \leq 40$ mm		$40 \text{ mm} < t \leq 80$ mm	
	f_y [N/mm ²]	F_u [N/mm ²]	f_y [N/mm ²]	F_u [N/mm ²]
EN 10025-2				
S 235	235	360	215	360
S 275	275	430	255	410
S 355	355	490	335	470
S 450	440	550	410	550

As restantes propriedades do material a adotar nos cálculos são também indicadas no EC3, e deverão tomar os seguintes valores:

- $E = 210$ GPa;
- $G = \frac{E}{2 \cdot (1 + \nu)} = 81$ GPa;
- $\nu = 0.3$
- $\alpha = 12 \cdot 10^{-6}$ por K

2.2.3. ANÁLISE GLOBAL

Segundo o EC3, os esforços poderão ser obtidos recorrendo a análises de primeira ordem ou de segunda ordem, estas últimas considerando os efeitos $P-\Delta$ através de processos sequenciais ou iterativos. Segundo a norma, a análise deverá incorporar os efeitos das tensões residuais e imperfeições geométricas, que podem ser tidas em consideração no pórtico através de uma imperfeição equivalente, com inclinação lateral das colunas na direção mais desfavorável, esta inclinação deverá ser calculada pela equação (2.1).

$$\phi = \phi_0 * \alpha_h * \alpha_m \quad (2.1)$$

Onde,

- ϕ_0 – valor de base: $\phi_0 = 1/200$;
- α_h – coeficiente de redução em função da altura: $\alpha_h = 2/\sqrt{h}$ com $2/3 \leq \alpha_h \leq 1.0$;
- α_m – coeficiente de redução associado ao número de colunas num piso, com um valor de esforço axial superior a 50% do valor médio por coluna no plano vertical, obtido por: $\alpha_m = \sqrt{0.5 * (1 + 1/m)}$, onde m é o número de colunas que verificam a condição.

2.2.4. CLASSIFICAÇÃO DAS SEÇÕES TRANSVERSAIS

O EC3 define quatro classes de seções transversais:

- Classe 1: permitem a formação de uma rótula plástica com capacidade de rotação suficiente para a aplicação dos métodos plásticos de análise;
- Classe 2: permitem atingir o momento resistente plástico mas as rotações são limitadas pela encurvadura lateral;
- Classe 3: permitem atingir a tensão de cedência mas poderão não atingir o momento plástico;
- Classe 4: cuja encurvadura local impede de atingir o momento de cedência.

Esta classificação depende da esbelteza dos componentes comprimidos e tem um papel muito significativo nas verificações de segurança, impondo restrições aos valores de tensão de cedência, aos valores de esforços axiais e momentos fletores resistentes e, também, ao tipo de análise que podem ser realizadas.

2.2.5. ESTADOS LIMITES DE UTILIZAÇÃO

Um estado limite de utilização (SLS) é um estado a partir do qual as condições de utilização da estrutura deixam ser verificadas. Embora possa ser especificado pelo dono de obra, em estruturas metálicas este estado limite normalmente é caracterizado em função da deformação das vigas para uma determinada combinação. Esta verificação é ilustrada na Fig. 2.1 e caracterizada pela equação (2.2), conforme ilustrado por Simões [6].

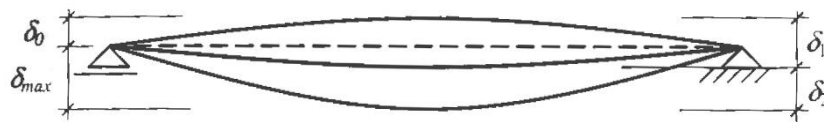


Fig. 2.1 - Deslocamentos verticais em vigas [6]

$$\delta_{max} = \delta_1 + \delta_2 - \delta_0 \quad (2.2)$$

Onde,

- δ_{max} – flecha no estado final em relação à linha reta que une os apoios;
- δ_0 – contra flecha da viga no estado não carregado;
- δ_1 – variação da flecha da viga devida às ações permanentes imediatamente após a sua aplicação;
- δ_2 – variação da flecha da viga devida à ação variável base associada aos valores de combinação das restantes ações variáveis.

2.2.6. ESTADO LIMITE ÚLTIMO

A segurança da estrutura deve ser verificada para cada secção, verificando-se a resistência aos esforços aplicados (N_{Ed} , V_{Ed} e M_{Ed}), e para cada elemento, verificando-se a resistência a fenómenos de encurvadura. A nível das verificações seccionais, estas são definidas na Secção 6.2 do EC3 e traduzem-se pelas equações (2.3) a (2.6), nas quais os valores resistentes ($N_{t,Rd}$, $N_{c,Rd}$, $V_{pl,Rd}$ e $M_{N,Rd}$) devem ser calculados em função da classe da secção.

$$\frac{N_{Ed}}{N_{t,Rd}} \leq 1.0 \quad (2.3)$$

$$\frac{N_{Ed}}{N_{c,Rd}} \leq 1.0 \quad (2.4)$$

$$\frac{M_{Ed}}{M_{N,Rd}} \leq 1.0 \quad (2.5)$$

$$V_{Ed} \leq 0.5 \times V_{pl,Rd} \quad (2.6)$$

As equações (2.3) e (2.4) verificam a resistência ao esforço axial atuante (N_{Ed}) em tração ou compressão, respetivamente. Pode verificar-se através da equação (2.5) se o valor de momento atuante não supera o valor do momento resistente em flexão composta, contemplando, se necessário, a influência do esforço axial atuante. Para secções de classe 1 e 2, este deverá ser tido em consideração sempre que as equações (2.7) não sejam verificadas. Nesse caso, para secções do tipo I ou H, o momento resistente deverá ser obtido pela equação (2.8).

$$\begin{cases} N_{Ed} \leq 0.25 * N_{pl,Rd} \\ N_{Ed} \leq \frac{0.5 * h_w * t_w * f_y}{\gamma_{M0}} \end{cases} \quad (2.7)$$

$$M_{N,y,Rd} = M_{pl,y,Rd} * (1 - n) / (1 - 0.5a) \text{ mas } M_{N,y,Rd} \leq M_{pl,y,Rd} \quad (2.8)$$

Em que:

- $n = N_{Ed} / N_{pl,Rd}$
- $a = (A - 2 * b * t_f) / A$ mas $a \leq 0.5$
- h_w – altura da alma
- t_w (t_f) – espessura da alma (banzo)
- A – área da secção

Para seções de classe 3 e 4, a tensão máxima deverá satisfazer o critério estabelecido na equação (2.9). O valor de cálculo da tensão longitudinal local atuante devida ao esforço normal ($\sigma_{x,Ed}$) e ao momento fletor deve ser menor que o valor da tensão de cedência do aço (f_y) dividida de um coeficiente parcial de segurança.

$$\sigma_{x,Ed} \leq \frac{f_y}{\gamma_{M0}} \quad (2.9)$$

Por fim, a equação (2.6) garante que a interação do esforço transversal atuante (V_{Ed}) com o momento fletor resistente pode ser desprezada, caso contrário, deverá adotar-se no cálculo do esforço transversal resistente, uma tensão de cedência reduzida:

$$(1 - \rho) * f_y \quad (2.10)$$

Em que $\rho = (2 * V_{Ed}/V_{pl,Rd} - 1)^2$.

As verificações de elementos à encurvadura está definida na Secção 6.3 do EC3, na qual são descritas as metodologias a seguir para elementos uniformemente comprimidos, uniformes em flexão, uniformes em flexão composta com compressão, o método geral e encurvadura lateral para elementos com rótulas plásticas. De seguida, apresentam-se as equações a verificar na aplicação do terceiro método supracitado:

$$\begin{cases} \frac{N_{Ed}}{\chi_y \frac{N_{Rk}}{\gamma_{M1}}} + k_{yy} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\chi_{LT} \frac{M_{y,Rk}}{\gamma_{M1}}} + k_{yz} \frac{M_{z,Ed} + \Delta M_{z,Ed}}{\frac{M_{z,Rk}}{\gamma_{M1}}} \leq 1.0 \\ \frac{N_{Ed}}{\chi_z \frac{N_{Rk}}{\gamma_{M1}}} + k_{zy} \frac{M_{y,Ed} + \Delta M_{y,Ed}}{\chi_{LT} \frac{M_{y,Rk}}{\gamma_{M1}}} + k_{zz} \frac{M_{z,Ed} + \Delta M_{z,Ed}}{\frac{M_{z,Rk}}{\gamma_{M1}}} \leq 1.0 \end{cases} \quad (2.11)$$

Onde,

- N_{Ed} , $M_{y,Ed}$, $M_{z,Ed}$ – valores de cálculo dos esforços atuantes ao longo do elemento;
- $\Delta M_{y,Ed}$, $\Delta M_{z,Ed}$ – momentos devido ao deslocamento do eixo neutro, nas secções de classe 4;
- χ_y , χ_z – coeficientes de redução devido à encurvadura por flexão;
- χ_{LT} – coeficiente de redução devido à encurvadura lateral;
- k_{yy} , k_{yz} , k_{zy} , k_{zz} – fatores de interação determinados usando o Método 2 (Austro-Alemão), definido no Anexo B do EC3.

2.3. EUROCÓDIGO 8

2.3.1. GENERALIDADES

A regulamentação sísmica europeia, Eurocódigo 8 (EC8), aplica-se às obras de engenharia civil em regiões sísmicas, à exceção de estruturas especiais, como centrais nucleares e grandes barragens. Este regulamento tem por finalidade assegurar, no caso de ocorrência de sismos, a preservação das vidas humanas, o controlo dos danos causados ao edifício e a operacionalidade de estruturas importantes para a proteção civil. As imposições presentes no EC8 deverão ser utilizadas de forma complementar às disposições dos outros Eurocódigos.

O EC8 introduz na Secção 1.5 alguns conceitos particulares, em relação aos restantes Eurocódigo, como:

- Coeficiente de comportamento: serve para reduzir as forças numa análise linear, de modo a ter em conta a resposta não linear da estrutura. É obtido em função do material, do tipo de sistema estrutural e dos procedimentos de projeto;
- Capacidade real (*capacity design*): método de cálculo no qual alguns elementos do sistema são escolhidos e projetados de modo a assegurar a dissipação de energia, enquanto os restantes elementos estruturais são dotados de resistência suficiente para manter o sistema seguro;
- Estrutura dissipativa: estrutura com capacidade de dissipar energia por meio de comportamento dúctil, ou outros mecanismos;
- Coeficiente de importância: coeficiente que reflete as consequências do colapso estrutural do edifício em termos de vidas humanas, da sua importância para a segurança pública e para a proteção civil imediatamente após o sismo;
- Elemento não estrutural: elemento que devido à falta de resistência ou modo de ligação não é considerado no projeto sísmico;
- Elementos sísmicos primários e secundários: os primeiros são considerados na análise sísmica, enquanto a contribuição dos segundos para a resistência lateral da estrutura é desprezada.

2.3.2. REQUISITOS DE DESEMPENHO E CRITÉRIOS DE CONFORMIDADE

Para cumprir os requisitos do regulamento, deverão ser satisfeitos os estados limites últimos e de limite de danos. Os primeiros são associados ao colapso ou formas de rotura que possam por em perigo a segurança das pessoas, e os segundos são associados ao controlo dos danos após os quais determinados requisitos de utilização deixam de ser satisfeitos.

Se possível, a estrutura deverá ter formas simples e regulares, tanto em planta como em altura, e, se necessário, estar dividida pelo meio de juntas em bloco dinamicamente independentes. Devem ser evitadas roturas frágeis, como mecanismos do tipo *soft storey*, recorrendo-se ao método do *capacity design*.

2.3.3. CARACTERIZAÇÃO DA AÇÃO SÍSMICA

O EC8 distingue dois tipos de ação sísmica: tipo 1 e tipo 2. A ação do tipo 1 corresponde a um sismo afastado, de magnitude superior a 5.5 (no caso de Portugal com epicentro no oceano Atlântico) e a ação do tipo 2 corresponde a um sismo próximo, de menor magnitude (no caso de Portugal com epicentro no território nacional).

A caracterização da ação sísmica está dependente maioritariamente de dois fatores: tipo de terreno (divididos em 5 tipos: A, B, C, D e E) e zona do território nacional (em Portugal dividido em 6 e 5 zonas, consoante o sismo seja do tipo 1 ou 2, respetivamente). Na escolha dos valores de aceleração à superfície (a_{gR}) é considerado como referência um sismo com um período de retorno de 50 anos.

A ação sísmica é traduzida por espectros de resposta elástica que traduzem a componente horizontal das acelerações (Fig. 2.2), onde as ordenadas representam a aceleração máxima a que um sistema de um grau de liberdade está sujeito durante um evento sísmico.

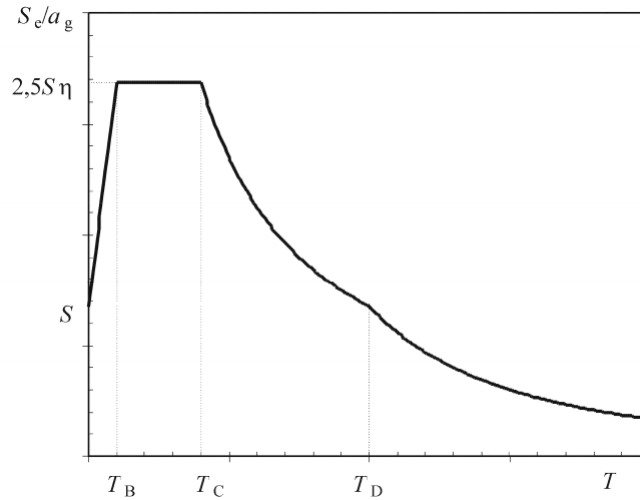


Fig. 2.2 - Espectro de resposta elástico [2]

O espectro é definido pelas seguintes expressões:

$$\begin{aligned}
 0 \leq T \leq T_B & \quad S_e(T) = a_g * S * \left[1 + \frac{T}{T_B} * (\eta * 2.5 - 1) \right] \\
 T_B \leq T \leq T_C & \quad S_e(T) = a_g * S * \eta * 2.5 \\
 T_C \leq T \leq T_D & \quad S_e(T) = a_g * S * \eta * 2.5 * \left[\frac{T_C}{T} \right] \\
 T_D \leq T \leq 4s & \quad S_e(T) = a_g * S * \eta * 2.5 * \left[\frac{T_C T_D}{T^2} \right]
 \end{aligned} \tag{2.12}$$

Onde,

- $S_e(T)$ – aceleração espectral;
- a_g – valor de cálculo da aceleração no terreno tipo A;
- T_B – limite inferior do período no patamar de aceleração espectral constante;
- T_C – limite superior do período no patamar de aceleração espectral constante;
- T_D – valor que define no espectro o início do ramo de deslocamento constante;

- S – coeficiente do solo;
- η – fator de correção do amortecimento

2.3.4. MÉTODO DE ANÁLISE POR FORÇAS LATERAIS EQUIVALENTES

A análise de estruturas pelo método das forças laterais equivalentes é um dos tipos de análise permitido pelo EC8. Este método pode ser aplicado a edifícios cuja resposta seja controlada pelo modo fundamental de vibração. Para validar esta condição, o EC8 estabelece que o período de vibração deverá ser, simultaneamente, inferior a $4 \cdot T_C$ e a 2 segundos, e satisfazer os critérios de regularidade em altura.

O valor da força de corte na base deverá ser calculado de acordo com a equação (2.13), na qual T_1 representa o período de vibração fundamental da estrutura, m a massa total do edifício acima da fundação ou cave rígida e λ um fator de correção em função da altura e do período de vibração do edifício.

$$F_b = S_d(T_1) * m * \lambda \quad (2.13)$$

Esta força deverá ser distribuída ao longo dos vários pisos em função dos deslocamentos destes no modo de vibração fundamental ou, caso estes não sejam conhecidos, em função da altura dos pisos. Para obter a distribuição das forças em função dos deslocamentos dos pisos deverá ser utilizada a equação (2.14), onde F_i corresponde à força horizontal atuante no piso i , e s_{ij} e m_{ij} aos deslocamentos e massas nos pisos i/j , respetivamente.

$$F_i = F_b * \frac{s_i * m_i}{\sum s_j * m_j} \quad (2.14)$$

2.3.5. VERIFICAÇÕES DE SEGURANÇA

Segundo o EC8, para verificar a segurança deverão ser verificados o estado limite último e a limitação de danos. No estado limite último devem ser verificadas condições de resistência e de ductilidade, nestas últimas sendo feita uma distinção entre as verificações de elementos dissipativos e de elementos não dissipativos. Na limitação de danos devem ser verificados os deslocamentos entre pisos sob uma ação sísmica com uma maior probabilidade de ocorrência que o ação sísmica de cálculo.

2.3.5.1. Estado limite último

Em conformidade com o objetivo de garantir que as rótulas plásticas se formam nas vigas ou nas ligações das vigas às colunas, e não nas colunas, o EC8 define um conjunto de verificações diferentes a realizar para as vigas e para as colunas. Assim, as vigas devem ser verificadas de acordo com as expressões (2.15) a (2.17), onde V_{Ed} deverá ser obtido somando o valor de cálculo das ações não sísmicas ao valor de cálculo do esforço transversal devido à aplicação dos momentos plásticos com sinais opostos nas extremidades da viga. Os valores das resistências de cálculo devem ser obtidos de acordo com o EC3.

$$\frac{M_{Ed}}{M_{pl,Rd}} \leq 1.0 \quad (2.15)$$

$$\frac{N_{Ed}}{N_{pl,Rd}} \leq 0.15 \quad (2.16)$$

$$\frac{V_{Ed}}{V_{pl,Rd}} \leq 0.5 \quad (2.17)$$

As colunas deverão ser calculadas em compressão considerando a combinação mais desfavorável do esforço normal e dos momentos fletores. À exceção do esforço transversal, que deverá satisfazer a equação (2.17), os restantes esforços deverão verificar a condição de resistência $E_d \leq R_d$. No entanto, para as colunas, os esforços de cálculo deverão ser obtidos conforme as equações (2.18) a (2.20). Nas quais os valores $N_{Ed,G}$, $M_{Ed,G}$ e $V_{Ed,G}$ deverão ser obtidos para as ações não sísmicas na combinação sísmica e os valores $N_{Ed,E}$, $M_{Ed,E}$ e $V_{Ed,E}$ deverão ser obtidos para as ações sísmicas de cálculo. γ_{ov} tem um valor recomendado de 1.25 e Ω é obtido segundo a equação (2.21), traduzindo-se pelo menor valor entre todas as vigas da estrutura.

$$N_{Ed} = N_{Ed,G} + 1.1 * \gamma_{ov} * \Omega * N_{Ed,E} \quad (2.18)$$

$$M_{Ed} = M_{Ed,G} + 1.1 * \gamma_{ov} * \Omega * M_{Ed,E} \quad (2.19)$$

$$V_{Ed} = V_{Ed,G} + 1.1 * \gamma_{ov} * \Omega * V_{Ed,E} \quad (2.20)$$

$$\Omega = \min \left(\frac{M_{pl,Rd,i}}{M_{Ed,i}} \right) \quad (2.21)$$

O cálculo do parâmetro Ω , contudo, tem sido alvo de discussão, pois poderá conduzir a valores consideravelmente subestimados para estruturas com cargas verticais preponderantes. Assim, Elghazouli [7] propõe a equação (2.22), como uma alternativa mais precisa de estimar este valor.

$$\Omega_{mod} = \min \left(\frac{M_{pl,Rd,i} - M_{Ed,G,i}}{M_{Ed,E,i}} \right) \quad (2.22)$$

2.3.5.2. Sensibilidade do deslocamento relativo entre pisos

O EC8 indica a equação (2.22) como critério a cumprir para evitar a realização de uma análise de segunda ordem. Não obstante, limita o valor de θ a 0.3, independentemente da análise realizada.

$$\theta = \frac{P_{tot} * d_r}{V_{tot} * h} \leq 0.10 \quad (2.23)$$

Onde,

- θ – coeficiente de sensibilidade de deslocamento relativo entre pisos;
- P_{tot} – carga gravítica total de todos os pisos acima do considerado, incluindo este;
- d_r – valor do deslocamento relativo entre pisos, avaliando a diferença entre os deslocamentos laterais no topo e na base do piso;
- V_{tot} – força de corte sísmica no piso considerado;
- h – altura entre pisos.

2.3.5.3. Condição de ductilidade local

Na sequência do previamente explanado na Secção 2.3.1., apresenta-se de seguida a fórmula proposta pelo EC8 com o objetivo de garantir uma adequada hierarquia de resistência entre os elementos de um pórtico, à exceção do nível superior do último piso.

$$\sum M_{RC} \geq 1.3 * \sum M_{Rb} \quad (2.24)$$

Onde,

- $\sum M_{RC}$ - soma dos valores de cálculo dos momentos resistentes dos pilares ligados ao nó, em função do esforço axial ao qual estão sujeitos;
- $\sum M_{Rb}$ - soma dos valores de cálculo dos momentos resistentes das vigas ligadas ao nó.

2.3.5.4. Limitação de danos

O regulamento considera satisfeito este requisito se, perante uma ação sísmica com probabilidade de ocorrência superior à ação sísmica de cálculo adotada para as verificações ao estado limite último, os deslocamentos entre pisos forem limitados de acordo com as equações (2.25) a (2.27), em função da ductilidade dos elementos não estruturais.

Onde, a equação (2.25) deverá ser utilizada para edifícios com elementos não estruturais constituídos por materiais frágeis fixos à estrutura, a equação (2.26) para edifícios com materiais não estruturais dúcteis e a equação (2.27) para edifícios cujos elementos não estruturais não tenham qualquer influência nas deformações estruturais.

$$d_r * v \leq 0.005 * h \quad (2.25)$$

$$d_r * v \leq 0.0075 * h \quad (2.26)$$

$$d_r * v \leq 0.010 * h \quad (2.27)$$

Na equação supracitada, d_r corresponde ao valor de deslocamento entre pisos (*drift*), h corresponde à altura entre pisos e v é um coeficiente de redução que, segundo o Anexo Nacional do EC8, assume o valor de 0.4 para ações sísmica do tipo 1 e 0.55 para ações sísmica do tipo 2.

2.3.6. COEFICIENTE DE COMPORTAMENTO - IMPROVED FORCE-BASED DESIGN

A semelhança de muitos regulamentos a nível mundial, por questões económicas e estéticas, o EC8 define que as estruturas deverão ser dimensionadas para uma resposta inferior à que seria de esperar em regime elástico, dissipando o excedente de energia através de comportamento plástico. Esta redução reflete-se no coeficiente de comportamento, q , um parâmetro empírico mas amplamente adotado pelas normas sísmicas em vigor a nível mundial. Assim, o EC8 define três classes de ductilidade: baixa (DCL), média (DCM) e alta (DCH); quanto maior a classe de ductilidade maior o coeficiente de comportamento permitido, contudo as restrições e imposições ao nível do detalhe construtivo e das verificações de segurança também aumentam.

O EC8 não apresenta sugestão sobre a forma de cálculo deste coeficiente numa situação de dimensionamento, apresentado apenas os valores máximos entre os quais este pode variar, pelo que o valor adotado é frequentemente o seu limite superior. Tomar esta opção, embora o cálculo seja consideravelmente mais expedito, poderá conduzir a soluções muito pouco otimizadas; pelo que, em 2009, foi proposta uma metodologia denominada *Improved Force-Based Design* (IFBD) [1] que procura ajustar o valor do coeficiente de comportamento à estrutura, dentro dos limites regulamentares, de modo a evitar o sobredimensionamento das estruturas. O processo proposto é o seguinte:

1. Determinação do corte basal elástico com base nas características dinâmicas da solução estrutural a testar;
2. Cálculo do coeficiente de comportamento da estrutura através da razão entre o corte basal elástico associado à solução, de acordo com o EC8, e o corte basal que resulta na formação da primeira rótula plástica;
3. Análise da estrutura para corte basal de dimensionamento associado ao coeficiente de comportamento obtido em 2;
4. Verificar se o coeficiente de sensibilidade (θ) se encontra dentro dos limites estabelecidos pelo EC8 e, seguidamente, verificar os elementos estruturais. Caso não verifique, a solução deverá ser ajustada e o processo repetido.

A Fig. 2.3 ilustra a relação entre o presente método e o EC8. Conforme se pode verificar, ao utilizar um valor de corte basal mais adequado no dimensionamento é possível reduzir significativamente o coeficiente de sobrerresistência (Ω). Como seria expectável, este aumento do controlo da resistência da estrutura, que se traduz no ajuste do coeficiente de comportamento, conduz a valores do coeficiente de sensibilidade muito inferiores.

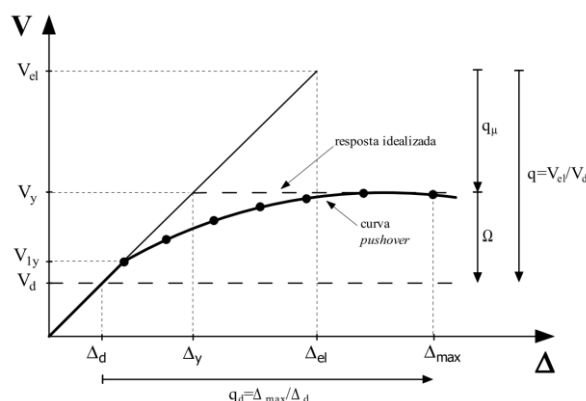


Fig. 2.3 - Curva de resposta lateral [1]

2.4. ALGORITMOS DE OTIMIZAÇÃO

A otimização consiste na procura da melhor solução dentro de um conjunto de combinações possíveis de variáveis. Esta solução é normalmente obtida maximizando ou minimizando uma função objetivo. No caso das estruturas metálicas, esta função objetivo é, normalmente, o peso total dos elementos estruturais, acrescido de uma penalização de peso em função dos elementos que não verifiquem as disposições regulamentares. Para problemas com poucos elementos este processo é bastante expedito, podendo ser resolvido manualmente com poucas tentativas. Contudo, à medida que as variáveis aumentam, a complexidade do problema e o número de combinações possíveis também aumentam, tornando o problema inadequado para resolver manualmente ou até por tentativa de todas as combinações possíveis (técnica *brute force*), com recurso a um computador.

Como resposta a problemas com formulações semelhantes, têm sido desenvolvidos algoritmos que resolvem de forma inteligente estes problemas de otimização. Existem algoritmos de dois tipos: os exatos, que garantem a melhor solução possível, embora o tempo de pesquisa possa ser irrazoável, e os algoritmos aproximados, que encontram a melhor solução para um intervalo de tempo definido pelo utilizador.

Os algoritmos meta-heurísticos são um tipo de algoritmos aproximados, frequentemente inspirados em fenómenos naturais, como a genética, o arrefecimento de metais, colónias de formigas ou abelhas, a pesquisa de uma harmonia, etc. Ao longo dos últimos anos, têm surgido vários artigos sobre a aplicação destes algoritmos a problemas de engenharia civil [8-14], com destaque para as estruturas metálicas, uma vez que, para a maioria dos problemas, as variáveis são discretas (listas de perfis possíveis).

2.4.1. PESQUISA HARMÓNICA

Os algoritmos de pesquisa harmónica (HS) são algoritmos meta-heurísticos inspirados nas bandas de Jazz, nas quais vários músicos tocam de improviso procurando encontrar uma harmonia entre si. Assim, um algoritmo deste tipo deve ser iniciado com a atribuição das características do músico, como a sua capacidade de memória, a tendência para afinar o instrumento, etc. Estes parâmetros de caracterização do comportamento são apresentados na Tabela 1 através de uma comparação entre as ações na criação de uma melodia e de uma estrutura.

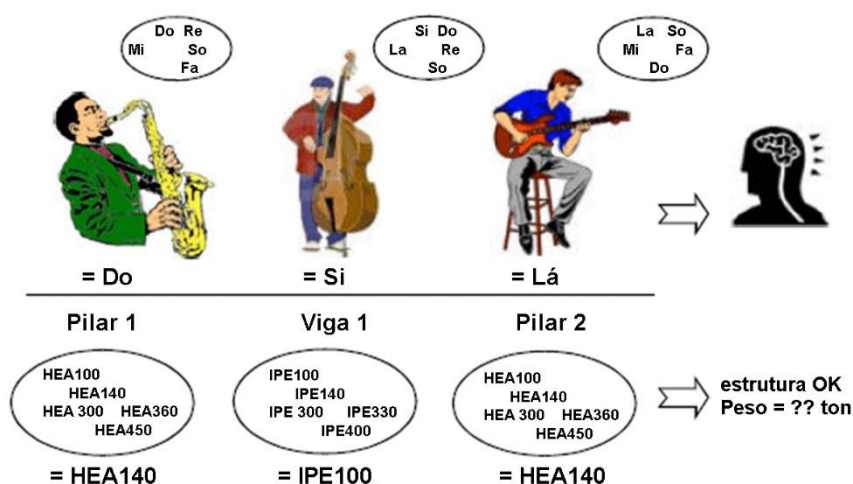


Fig. 2.4 - Analogia entre uma banda de Jazz e um problema de dimensionamento

Tabela 2 - Analogia dos parâmetros de caracterização da pesquisa harmônica

Parâmetro	Descrição	Músico	Estrutura
HMS	Harmonic Memory Size	Quantidade de melodias agradáveis guardadas na memória	Número de estruturas, melhor classificadas, a guardar
PAR	Pitch Adjustment Rate	Frequência com que afina o instrumento	Probabilidade de afinar um determinado elemento
BW/MPAI	Bandwidth/ Maximum Pitch Adjustment Index	Intervalo de afinações entre as quais o músico varia	Número de perfis que o elemento poderá subir ou descer
HMCR	Harmony Memory Consideration Rate	Probabilidade de tocar uma melodia de memória em vez de improvisar	Probabilidade de usar como ponto de partida uma estrutura da memória

Embora na versão original tenha sido proposto um processo no qual estes parâmetros eram constantes, tem-se verificado que para alguns problemas resulta numa convergência mais tardia. Como resultado, têm vindo a ser desenvolvidos vários trabalhos que propõem um ajuste dinâmico dos parâmetros em função do progresso da pesquisa harmônica. Recentemente, foi proposto por Kumar, *et al.* [11] uma versão de *Parameter Adaptive Harmony Search* que ajusta dinamicamente estes parâmetros de forma a, numa fase inicial permitir uma grande variedade de soluções explorando todo o campo de possibilidades, e ir convergindo para a melhoria das soluções na memória à medida que o processo avança. Este processo utiliza uma variação exponencial do parâmetro BW, proposta anteriormente por Mahdavi, *et al.* [12], equação (2.28), e uma variação linear do parâmetro HMCR, equação (2.29), e uma variação exponencial do parâmetro PAR, equação (2.30), proposta pelo autor.

$$BW(gn) = BW_{max} * e^{\left(\frac{\ln(\frac{BW_{min}}{BW_{max}})}{NI} * gn\right)} \quad (2.28)$$

$$HMCR(gn) = HMCR_{max} * \frac{(HMCR_{max} - HMCR_{min})}{NI} * gn \quad (2.29)$$

$$PAR(gn) = PAR_{max} * e^{\left(\frac{\ln(\frac{PAR_{min}}{PAR_{max}})}{NI} * gn\right)} \quad (2.30)$$

Onde, as variáveis com índice “max” ou “min” se referem aos intervalos de valores possíveis para as respectivas variáveis, NI representa o número máximo de iterações e gn representa a iteração atual.

Assim como os parâmetros de iniciação, a definição da função objetivo é crucial para um funcionamento eficiente do algoritmo. Esta é definida pelo peso dos elementos adicionado de uma penalização, pelo que torna a sua definição bastante complexa, pois as penalizações poderão desequilibrar o algoritmo caso tenham um peso desadequado. Neste trabalho, a função objetivo será definida tendo por base as funções utilizadas por Pezeshk, *et al.* [13] e Camp, *et al.* [8] para algoritmos genéticos e de colónias de formigas, respetivamente, assim como, mais recentemente, por Degertekin [9], para algoritmos de pesquisa harmónica. De seguida, são apresentadas as expressões utilizadas nesta última publicação:

$$\text{Minimize } \varphi(x) = W(x)[1 + C]^\varepsilon \quad (2.31)$$

$W(x)$ representa o peso total da estrutura, C o resultado da função correspondente às violações de critérios e ε o expoente da função de penalização, assumindo o valor de 2 na publicação de Camp, *et al.* [8].

$$C = \sum_{i=1}^{N_S} C_I^d + \sum_{i=1}^{N_C} C_I^l \quad (2.32)$$

Em que,

- C – somatório das razões entre os valores verificados e os permitidos para cada elemento, em estado limite de serviço e de estado limite último para ações gravíticas;
- C_I^d – razão entre os deslocamentos verificados e os permitidos
- C_I^l – razão entre os valores de esforços atuantes e os resistentes
- N_S – número de pisos
- N_C – número de pilares e vigas

Por fim, apresenta-se um esquema de funcionamento do algoritmo, com o objetivo de ilustrar os processos associados ao mesmo.

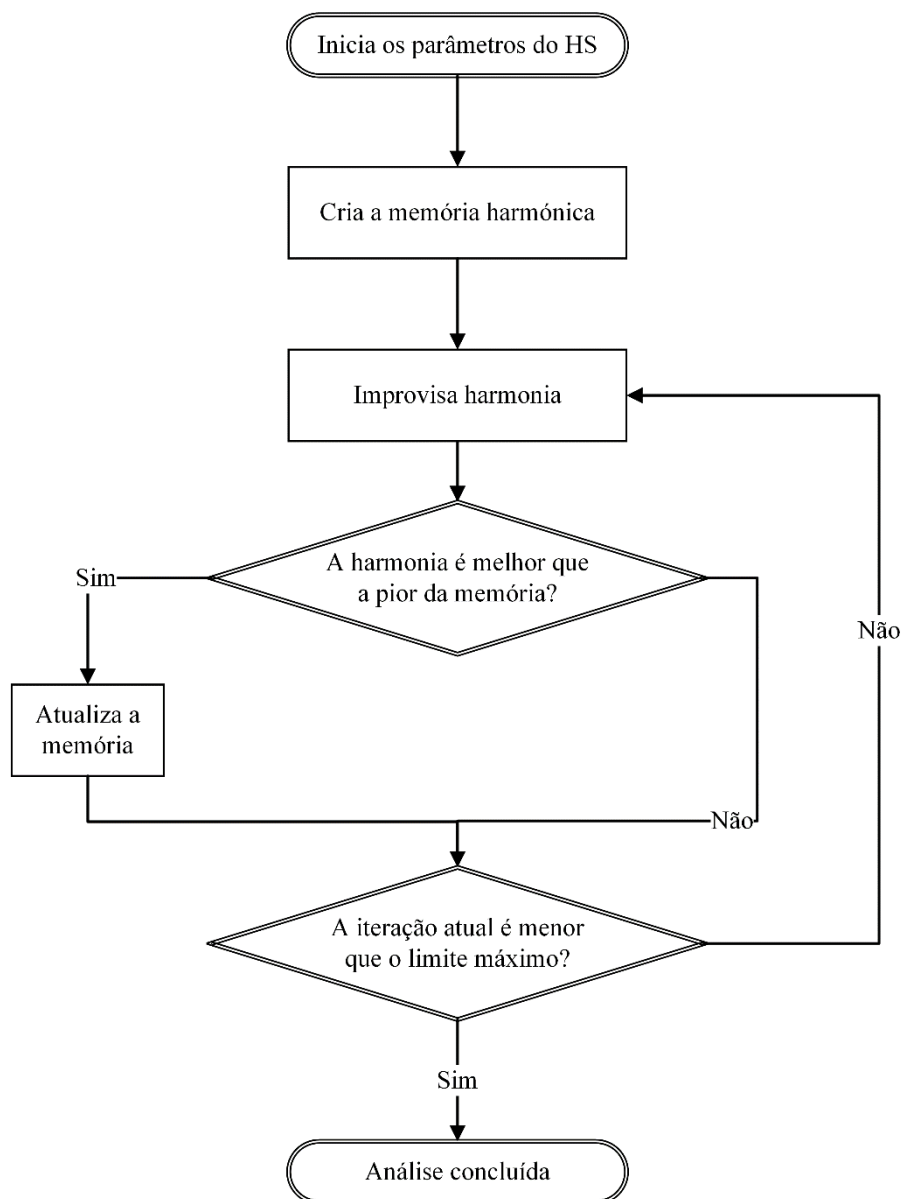


Fig. 2.5 - Esquema de funcionamento da pesquisa harmônica

2.5. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as regras gerais e prescrições que deverão ser utilizadas no dimensionamento de pórticos metálicos, segundo as normas europeias. Foi, também, introduzido o conceito de algoritmo de otimização e explicado o funcionamento de um algoritmo de pesquisa harmônica.

No próximo capítulo será apresentado o processo de conceptualização e desenvolvimento da ferramenta informática de cálculo otimizado de estruturas.

3

DESENVOLVIMENTO DE UMA FERRAMENTA PARA CÁLCULO OTIMIZADO DE ESTRUTURAS METÁLICAS

3.1. DEFINIÇÃO DOS OBJETIVOS

Este trabalho tem como principal objetivo o desenvolvimento de uma ferramenta informática de cálculo estrutural que efetue o dimensionamento de pórticos metálicos para ações gravíticas e sísmicas, e que seja integrável num processo de otimização, de modo a ser possível obter uma estrutura otimizada num intervalo de tempo razoável. Numa fase inicial, procurou-se desenvolver uma ferramenta capaz de abranger todos os parâmetros do dimensionamento gravítico e sísmico de uma estrutura metálica, em conformidade com as regras definidas nos Eurocódigos 3 [4] e 8 [2]. Contudo, houve a preocupação de garantir a expansibilidade da ferramenta, sendo um dos pilares desta expansibilidade o *software* utilizado para realizar as análises estruturais, o OpenSees [3], que será alvo de referência, posteriormente, neste no documento.

Pretende-se que a ferramenta seja livre, pelo que se recorreu apenas a *software Open Source* para a suportar. Todo o código foi desenvolvido em Python 2.7.9 e as análises realizadas em OpenSees 2.4.5, sendo a única exceção o ficheiro de *input*, criado com recurso ao Excel (formato *xls*) uma vez que este é um *software* amplamente usado na engenharia civil e com alternativas viáveis para a criação do ficheiro, como o LibreOffice.

3.2. MODELO DE INFORMAÇÃO

A ferramenta de cálculo resultou de um longo processo de discussão de variáveis e possíveis ramificações, tendo sido necessário elaborar um modelo de informação complexo o suficiente para cumprir com os objetivos estabelecidos para este trabalho. Os pontos-chave desta conceptualização e a estruturação do modelo de informação serão apresentados ao longo deste subcapítulo.

3.2.1. INTRODUÇÃO

No desenvolvimento da ferramenta procurou-se conceber um modelo de informação eficiente e modular, ou seja, que permitisse uma boa interoperabilidade entre as várias funções necessárias e ainda a adição de novas funcionalidades sem comprometer o funcionamento do código existente. Adicionalmente, este

modelo de informação teria de ser desenvolvido em função das capacidades do Python e da sua estrutura enquanto linguagem de programação orientada por objetos. Assim, uma representação do modelo implementado no Python é apresentado na Fig. 3.1.

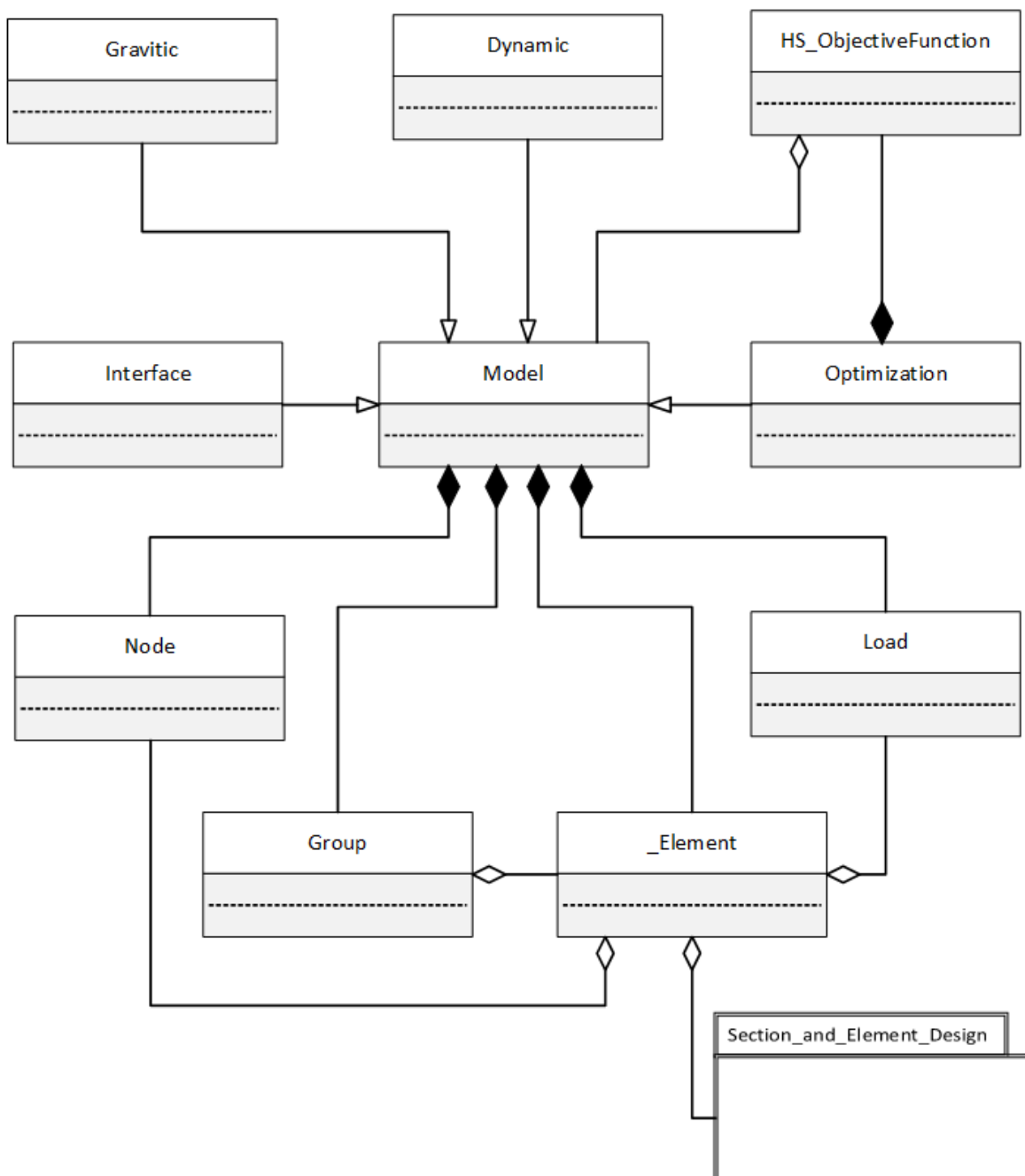


Fig. 3.1 - Esquema global de funcionamento do programa

O modelo elaborado é constituído por um objeto central, o modelo, que agrega informação geométrica (nós, barras, condições de apoio), as secções possíveis, resultados de análises, etc. e tem acesso a funções que realizam as várias análises e as correspondentes verificações de segurança, algoritmos de otimização e funções de interface.

No desenvolvimento deste trabalho, preparou-se uma interface de introdução de dados capaz de receber informação sobre:

- a geometria da estrutura e grupos de elementos;
- as propriedades dos materiais;
- os casos de cargas e valores das mesmas (para ação gravítica e vento);
- as variáveis que definem a ação sísmica, que virá a ser considerada segundo o método das forças laterais equivalentes.

Não obstante, o modelo de informação está preparado para que seja facilmente implementado um módulo de cálculo da ação do vento e um módulo de análise sísmica modal por espectro de resposta, entre outros que possam ser considerados de interesse.

3.2.2. INTRODUÇÃO DE DADOS

A forma como o *input* é realizado é um fator condicionante, pois determina a usabilidade da ferramenta. Assim, este foi alvo de cuidada atenção, tendo-se concluído que deveria ser dividido em 5 separadores: geometria e propriedades gerais, ações de elemento, ações nos nós, ações sísmicas e gamas de perfis a utilizar.

3.2.2.1. Geometria e Propriedades Gerais

Relativamente à geometria, optou-se por apenas permitir estruturas com alinhamentos regulares entre pisos sendo introduzido os valores, em metros, de cada vão e piso individualmente. Foi, contudo, implementada a possibilidade de pisos recuados em relação ao alinhamento da direita. Quanto aos materiais é permitido a introdução do módulo de elasticidade, coeficiente de *Poisson* e peso volúmico (valores únicos para toda a estrutura) e a distinção do valor da tensão de cedência para vigas e pilares. É, também, possível designar o interesse em considerar o peso próprio dos elementos, assim como a condição de simetria. Por fim, devem ser especificados os valores dos coeficientes de combinação desejados para as várias combinações de ações e o algoritmo de otimização que se pretende utilizar. A folha de Excel na qual é realizada esta introdução de dados é ilustrada na Fig. 3.2.

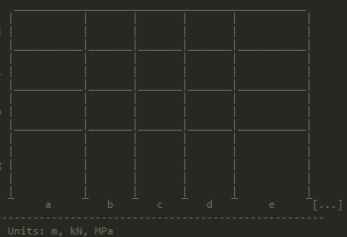
	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T					
1	[...]							ID	1															
2								Operating System												FEUP_Gri				
3								Materials																
4								E	210 MPa		Type of optimization	Harmony Search												
5								Fy_beam	275 MPa		No. of cores	1												
6								Fy_column	275 MPa		Definition of groups	Manual												
7								v	0,3															
8								h	77 kN/m3															
9								p																
10								Self_Weight												No	Manual Masses	Yes		
11								Design Condition												Symmetrical				
12																								
13																								
14																								
15																								
16																								
17	Geometri:1st		2nd	3rd	4th																			
18	L_Vaos	6	6	6																				
19	H_Pisos	4,5	3,5	3,5																				
20	Massas (O	45,72	45,72	34,2																				
21																								
22	Loads		γ (ELU)		ψ,0	ψ,1	ψ,2													setbacks?	No			
23	Case 1	Permanent	1,35	-	-	-		Types:	1 Distributed Load															
24	Case 2	Variable	1,5	0,7	0,5	0,3	2 Point load (mid span)																	
25	Case 3	Wind	1,5	0,6	0,2	0	3 Point load (2x 1/3 span)																	
26	Case 4	Seismic																						

Fig. 3.2 - Screenshot da folha de Excel para introdução dos dados gerais

3.2.2.2. Ações

A definição das ações foi implementada procurando um equilíbrio entre a versatilidade de opções pretendidas e as limitações e dificuldade inerentes ao OpenSees. Assim, concluiu-se que seria interessante uma solução na qual se permitia 4 tipos de carregamento: carga vertical uniformemente distribuída, carga vertical pontual a meio vão, cargas verticais pontuais aos terços e cargas verticais e/ou horizontais pontual nos nós da estrutura. Este último tipo de carga é introduzido num separador diferente com o objetivo de facilitar a introdução dos dados. Um excerto da folha de Excel utilizada na introdução das cargas é apresentado na Fig. 3.3.

	B	C	D	E	F	G	H	I	J	K	L	M
1	Floor	Load Case	Span 1		Span 2		Span 3		Span 4		Span 5	
2			Type	Value	Type	Value	Type	Value	Type	Value	Type	Value
3	1	1	2	-103,5	2	-103,5	2	-103,5				
4	2	1	2	-103,5	2	-103,5	2	-103,5				
5	3	1	2	-85,5	2	-85,5	2	-85,5				
6	1	2	2	-36	2	-36	2	-36				
7	2	2	2	-36	2	-36	2	-36				
8	3	2	2	-18	2	-18	2	-18				
9												
10												
11												
12												
13												
14												
15												

Fig. 3.3 - Excerto da folha de Excel utilizada para introdução de ações nos elementos

3.2.2.3. Ação Sísmica

As ações sísmicas (tipo 1 e tipo 2) são definidas através dos parâmetros de caracterização do EC8, ou seja, a zona onde o pórtico se localiza, o tipo de solo, a classe de importância, o amortecimento, a ductilidade, o parâmetro opcional relativo à ductilidade dos elementos não estruturais e a indicação do coeficiente de comportamento pretendido (valores de referencia do EC8, IFBD ou valor específico). Os valores de acelerações e períodos indicados pelo EC8 são obtidos na folha Excel (Fig. 3.4), de modo a dar ao utilizador a possibilidade de os editar, caso seja do seu interesse.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2					PT NA	No							
3	Zone	Lisboa				S.Zone	a_{gR} (m/s ²)	S_{max}	T_b (s)	T_c (s)	T_d (s)	γ_I	
4	Type of soil	B			Type 1	1.3	4,905	1,20	0,15	0,50	2,0	1,00	
5	Class of importance	II			Type 2	2.3	4,905	1,35	0,05	0,25	1,2	1,00	
6	Damping (%)	5											
7	Ductility	DCH											
8	Interstory Drift	0,0075											
9	Behaviour factor	EC8											
10													

Fig. 3.4 - Excerto da folha de Excel utilizada para introdução de ações sísmicas

3.2.2.4. Perfis a utilizar

No dimensionamento, o programa segue duas listas de perfis para dimensionar as seções, uma para vigas e outra para colunas. Para situações em que sejam utilizadas gamas completas com os vários perfis ordenados por área estas listas podem ser modificadas de forma simples utilizando a folha de Excel na qual se realiza o *input*. Contudo, optou-se por estruturar o programa de forma a permitir uma edição manual destas listas com relativa facilidade, por forma a permitir uma flexibilidade tão grande quanto desejada.

3.2.2.5. Grupos de pilares/vigas

Opcionalmente, poderão ser definidos grupos de elementos, garantindo que a todos os elementos do mesmo grupo será atribuída a mesma secção. Esta definição está implementada em folhas diferentes para pilares e para vigas por uma questão de organização da informação. Com o objetivo de ilustrar o funcionamento, na Fig. 3.5 é apresentado um excerto da folha de Excel utilizada para definir os grupos de pilares.

	B	C	D	E	F	G	H
1	Floor	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
2	1	1	2	2	1		
3	2	1	2	2	1		
4	3	3	4	4	3		
5	4	3	4	4	3		
6	5	5	6	6	5		
7							
8							

Fig. 3.5 - Excerto da folha de Excel utilizada para a definição dos grupos de pilares

3.2.3. GERAÇÃO DO MODELO

A geração do modelo teve como principal preocupação a interoperabilidade entre a ferramenta que estava a ser desenvolvida e o *software* que efetua a análise estrutural, duas entidades que manipulam os dados de forma completamente diferente, uma vez que realizam tarefas complementares. Assim, foi necessário desenvolver um sistema de identificação que permitisse identificar um elemento, inequivocamente, em ambos os sistemas. Para o efeito, elaborou-se um sistema de identificação em que a cada elemento e nó está associado um código com o seguinte formato: ABBCC, onde:

- A - assume os valores de 1, 2 e 3, consoante corresponda a viga, pilar ou nó, respetivamente;
- BB - identifica o andar, desde o piso 00 ao 99;
- CC - identifica a posição, desde o alinhamento 00 ao 99.

Assim, e a título de exemplo, o elemento com o código 10301 seria a viga do primeiro vão, no terceiro piso, e o elemento com o código 20203 seria o terceiro pilar do segundo andar. A Fig. 3.6 representa este sistema de numeração aplicado a uma pequena estrutura, a fim de ilustrar melhor o seu funcionamento.

Para implementar corretamente este sistema foi necessário criar, no programa, algumas classes de suporte. Assim, foram criadas a classe *Node* e *_Element*, onde para cada nó ou barra é criado um objeto da classe *Node* ou *_Element*, respetivamente. Este objeto, entre outras propriedades, tem o código do elemento. É pertinente referir, também, que durante a geração do modelo as cargas são atribuídos a objetos da classe *Load*, nos quais não há distinção entre elemento e nó, que têm como propriedades o código de identificação e as cargas correspondentes.

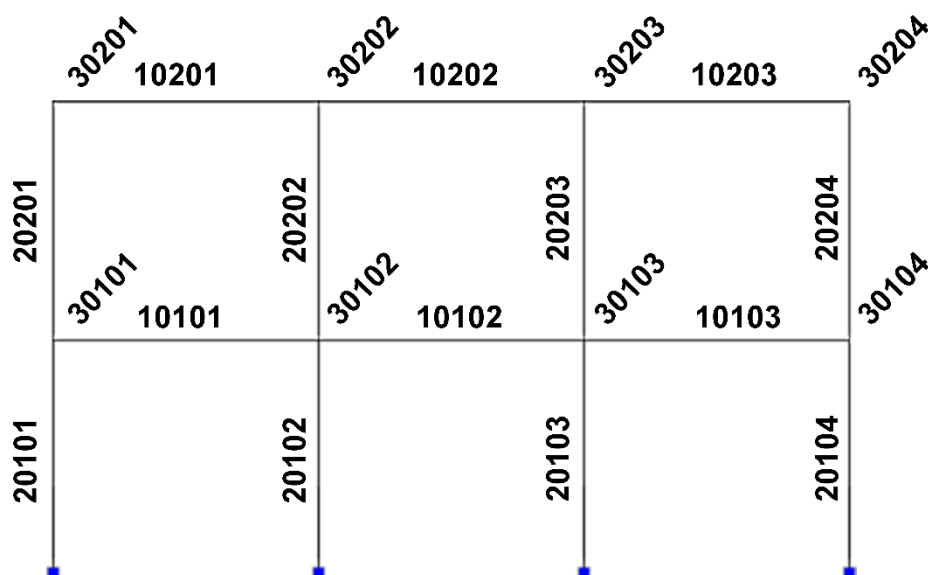


Fig. 3.6 - Exemplificação do sistema de catalogação

3.2.3.1. Pisos recuados

Tal como referido anteriormente, podem ser definidos pisos recuados (*setbacks*) na estrutura aquando da criação do ficheiro de *input*. Tendo em conta que a situação mais corrente será estruturas regulares em altura, e que a definição da geometria apenas representa uma ínfima porção do processo de procura da solução ótima, optou-se por, independentemente da existência de *setbacks*, criar uma estrutura regular em altura e, posteriormente, excluir do modelo de análise os nós e os elementos abrangidos pelos recuos.

3.3. ANÁLISES ESTRUTURAIS

Todas as análises são realizadas tendo em conta os efeitos $P-\Delta$ da estrutura, em conformidade com a Secção 5.2.1 do Eurocódigo 3 [4], como alternativa ao método mais frequentemente utilizado no qual se calcula o α_{cr} da estrutura. Com a finalidade de dar expressão à análise de segunda ordem, as imperfeições geométricas são considerados de acordo com a Secção 5.3.2 do EC3, através da aplicação dos valores das imperfeições geométricas propostos às coordenadas dos nós da estrutura inicial.

3.3.1. OPENSEES

O OpenSees é um dos *softwares Open Source* de simulação sísmica utilizando elementos finitos mais utilizados da atualidade, desenvolvido na Universidade da Califórnia em Berkeley, e com participação comunitária a nível mundial. A sua estrutura modular, representada na Fig. 3.7, potencia o seu

desenvolvimento, permitindo a adição de novas funcionalidade sem afetar a plataforma já existente. Para suportar esta filosofia de estruturação do *software*, a interface é minimalista, não havendo interface gráfica de introdução de dados; este fator, embora torne o OpenSees pouco *user-friendly* para modelação manual de estruturas, é uma mais-valia quando se procura uma modelação computadorizada destas.

O OpenSees é abrangente ao ponto de permitir a descrição dos componentes a três níveis: a nível do elemento, a nível da secção e a nível de cada fibra da secção. E, o facto de não ser uma “caixa negra” permite a introdução de novos materiais, secções, etc. com a garantia de controlo sobre o seu comportamento. Embora neste trabalho apenas tenham sido utilizadas análises estáticas, com a consideração de efeitos P- Δ , elementos elásticos e com a definição das massas por piso controlada pelo script de Python, o OpenSees tem um vasto leque de possibilidades de análises e definição das propriedades da estrutura que potencia amplamente a ferramenta.

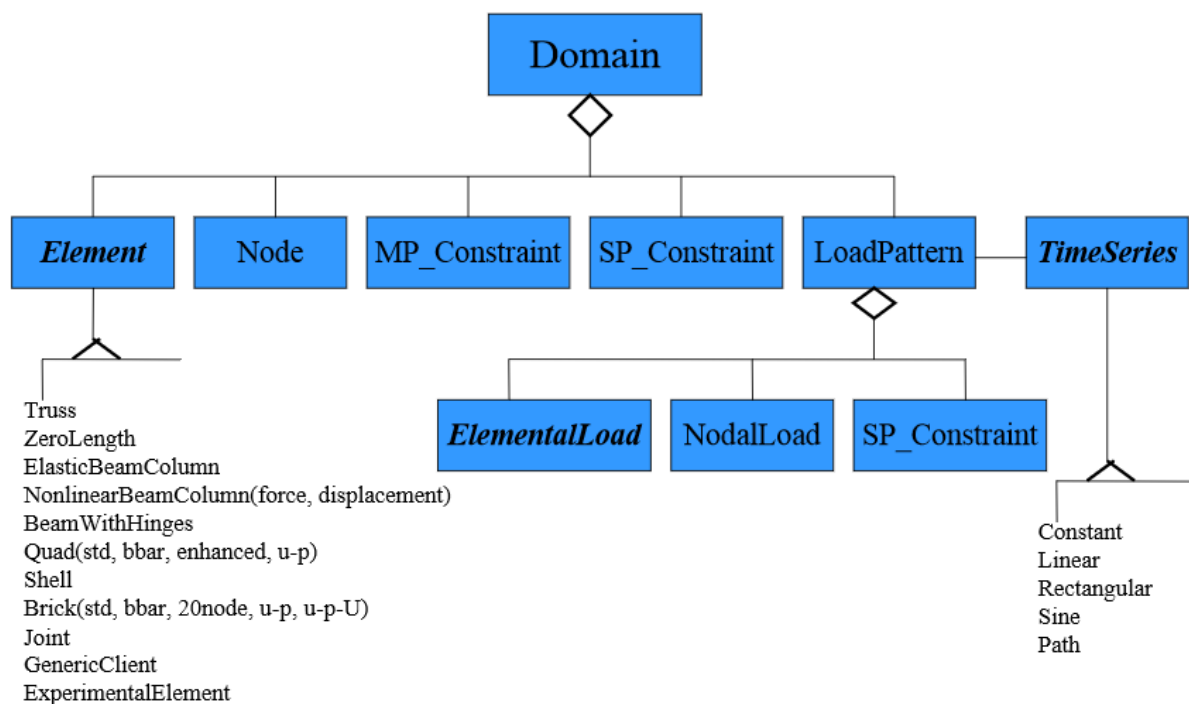


Fig. 3.7 - Estrutura modular do OpenSees [15]

A título de exemplo, refere-se que as simulações suportadas pelo OpenSees incluem análises *push-over*, *reversed-cyclic*, *dynamic time-series*, *uniform-support excitation* e *multi-support excitation*, sendo possível definir materiais com comportamentos plásticos, as suas leis de comportamento e critérios de degradação. A nível da secção é possível definir secções mistas, especificando as relações de comportamentos entre os dois materiais. A nível do elemento o comportamento pode ser definido em função do deslocamento ou da força, nós de ligação, entre dezenas de outras funcionalidades.

3.3.2. CRIAÇÃO DOS MODELOS DE ANÁLISE

Os ficheiros de modelação da estrutura são gerados através de *templates*, recorrendo ao Jinja 2, um módulo de Python. Estes *templates* têm definido, à partida, que o modelo deverá ser de duas dimensões,

com encastramentos nos nós à cota zero e a transformação geométrica do tipo *PDelta*. As cargas são criadas com a configuração: *pattern Plain # Linear*, onde as ações são introduzidas em conformidade com o definido no *input* e o peso próprio é traduzido por uma carga distribuída ao longo dos elementos.

3.3.3. ANÁLISE DINÂMICA

O cálculo das propriedades dinâmicas da estrutura é realizado no OpenSees, do qual se extrai os períodos, frequências e *eigenvalues* para os vários modos de vibração. Nesta análise optou-se por permitir ao utilizador especificar as massas por piso ou permitir ao programa defini-las em função da combinação “G” + $\psi_2 \cdot Q$.

À data da realização deste trabalho, para análise sísmica, apenas está implementado o método das forças horizontais equivalentes, pelo que o programa verifica a possibilidade de aplicação do mesmo com base nos critérios definidos na cláusula 4.3.3.2.1 do EC8 e, caso esta não seja verificada, emite um aviso. O corte basal na base é obtido pela equação (2.13) e é distribuído em altura no alinhamento mais à esquerda, através da equação (2.14).

3.3.4. COMBINAÇÕES

Como este projeto se foca nas cargas gravíticas, sísmicas e de vento, são geradas e armazenadas as seguintes combinações de dimensionamento, onde as primeiras 4 são usadas no dimensionamento gravítico, a oitava é utilizada de forma versátil para várias combinações voláteis e as restantes são usadas no dimensionamento sísmico. As combinações são as seguintes:

Tabela 3 - Combinações de dimensionamento geradas pelo programa

Combinação	Descrição	Ações envolvidas
1	Ação base: Sobrecarga (SLS)	“G” + “Q _s ” + ψ_0 “Q _v ”
2	Ação base: Sobrecarga (STR)	γ_G “G” + γ_{Qs} “Q _s ” + $\gamma_{Qv} \psi_0$ “Q _v ”
3	Ação base: Vento (SLS)	“G” + “Q _v ” + ψ_0 “Q _s ”
4	Ação base: Vento (STR)	γ_G “G” + γ_{Qv} “Q _v ” + $\gamma_{Qs} \psi_0$ “Q _s ”
5	Ação base: Sismo (estado limite último) - elementos dissipativos	“G” + ψ_2 “Q _s ” + “E”
6	Ação base: Sismo (estado limite último) - elementos não dissipativos	“G” + ψ_2 “Q _s ” + 1.1 $\gamma_{ov} \Omega$ “E”
7	Ação base: Sismo (estado limite de danos)	“G” + ψ_2 “Q _s ” + ν “E”.q
8	Combinação auxiliar para cálculos intercalares	
9	Ação base: Sismo (parcela gravítica)	“G” + ψ_2 “Q _s ”
10	Ação base: Sismo (parcela sísmica)	“E”

Note-se que existem várias combinações para a ação sísmica. Isto deve-se ao facto de o EC8 exigir que as verificações sejam realizadas para várias combinações de ações consoante o tipo de elemento em causa. Uma vez que se optou por realizar todas as análises considerando os efeitos P- Δ , não seria

possível sobrepor e escalar as ações. Assim foi necessário a implementação de uma combinação para a verificação de vigas (5), pilares (6) e estado de serviço (7). Nesta última multiplica-se a ação sísmica pelo coeficiente de comportamento para obter os deslocamentos reais e por ν para converter as acelerações sísmicas a considerar em estado limite último para estado de limite de limitação de danos. As combinações 9 e 10 são utilizadas para obter o coeficiente de sensibilidade ao deslocamento relativo entre pisos.

A influência que a sequência do carregamento tinha na análise também foi estudada, tendo sido consideradas as seguintes possibilidades: 1) todas as ações aplicadas em simultâneo, 2) estado inicial com as ações permanentes e, de seguida, as ações variáveis. Não se verificando alterações relevantes nos resultados, entre as duas abordagens.

3.3.5. PROCESSAMENTO E ARMAZENAMENTO DOS RESULTADOS

Os resultados obtidos das várias análises realizadas em OpenSees são, posteriormente, lidos pelo programa e atribuídos aos respetivos objetos. Assim, os deslocamentos nodais são atribuídos aos respetivos nós e os esforços nas extremidades aos respetivos elementos. Por fim, para os elementos são calculados os valores dos esforços em 5 seções para os pilares e em 13 seções para a viga, de forma a discretizar o elemento e a permitir a realização das várias verificações de segurança previstas nos Eurocódigos 3 e 8.

3.4. VERIFICAÇÕES DE SEGURANÇA

Para a realização das verificações foi idealizado um módulo independente que, para uma determinada secção e em função de um determinado conjunto de esforços, realizasse a verificação da segurança da secção no âmbito de um determinado regulamento. Este módulo foi alvo de discussão pois procurou-se desenvolver uma estrutura global que permitisse a implementação de todos os tipos de secções utilizadas na engenharia civil, embora, para este trabalho, apenas tenham sido implementadas as verificações relativas a secções metálicas, em I ou H, com momentos segundo o eixo principal. Este será explicado com mais detalhe na Secção 3.4.2.

A ponte entre o programa principal e o pacote supracitado é realizada através da classe *_Element*, que possui as informações das secções e dos esforços, na qual são preparados os dados para a criação das várias secções e do próprio elemento no pacote, e subsequente análise.

3.4.1. ESTADO LIMITE DE SERVIÇO

O estado limite de serviço é verificado para as combinações 1 e 3 da Tabela 3, em função de um limite máximo de flecha ($L/250$). Uma vez que o OpenSees não permite obter deslocamentos ao longo de um elemento, apenas nas suas extremidades, e tendo em conta que poderiam ser conjugadas cargas distribuídas com cargas pontuais, a obtenção do diagrama de momentos não seria direta para todos os casos. A solução proposta para a resolução deste problema foi recorrer ao princípio dos trabalhos virtuais, obtendo uma aproximação do diagrama de momentos através de uma regressão de quarto grau dos 13 pontos de momentos flectores conhecidos na viga.

A verificação no âmbito do EC8 é realizada através da verificação da equação (2.27), para a combinação 7 da Tabela 3, em função do critério para o coeficiente de comportamento definido pelo utilizador.

3.4.2. ESTADO LIMITE ÚLTIMO

3.4.2.1. Verificações seccionais

As verificações seccionais, no âmbito do EC3, são realizadas em conformidade com as indicações prescritas para elementos em flexão composta, conforme referido na Secção 2.2.6 do presente trabalho, com recurso ao módulo *section_and_element_design*.

As vigas são verificadas, no âmbito do EC8, de acordo com o definido para as equações (2.15) a (2.17). Considerando que as análises têm em conta os efeitos $P-\Delta$, a verificação dos pilares é realizada para uma combinação de esforços cujo cálculo, em Opensees, é realizado para a combinação 6 da Tabela 3, por oposição à sobreposição de efeitos sugerida no EC8.

A verificação associada ao *Capacity Design* é feita através de uma função independente que calcula os momentos resistentes dos pilares e das vigas, tendo em conta eventuais interações com o esforço axial, para a combinação 5 da Tabela 3, e verifica a relação de momentos em todos os nós à exceção dos nós da cobertura, em conformidade com o ponto 4.4.2.3(6) do EC8.

3.4.2.2. Verificações à encurvadura

Por uma questão de uniformização de processos, considerou-se a verificação de todos os elementos num regime de flexão composta com compressão, através da verificação das condições indicadas em (2.11). Conforme indicado na norma, foi considerado para todos os elementos não dissipativos que o valor do comprimento de encurvadura era igual ao valor do comprimento real do elemento.

3.4.3. DESENVOLVIMENTO DO MÓDULO DE VERIFICAÇÃO DE SEGURANÇA

Para a verificação das seções e dos elementos optou-se por desenvolver um módulo que funcionasse de forma completamente independente do programa base. Uma vez que a utilidade de um conjunto de funções de análise de seções excede amplamente este trabalho, optou-se por estruturar um módulo que pudesse ser transversal a todo o tipo de seções e normas regulamentares. Assim, embora no âmbito deste trabalho apenas tenham sido desenvolvidas as funções relativas a seções e elementos metálicos, na Fig. 3.8 é apresentada a estrutura idealizada que permitirá, em trabalhos futuros, estender o módulo a outras áreas da engenharia civil. Foram, também, implementadas outras funcionalidades como cálculo de curvas de interação. O desenvolvimento deste módulo usou como referência um conjunto de ferramentas desenvolvidas num trabalho anterior por Barros [16], contudo foi completamente reformulado e é independente deste.

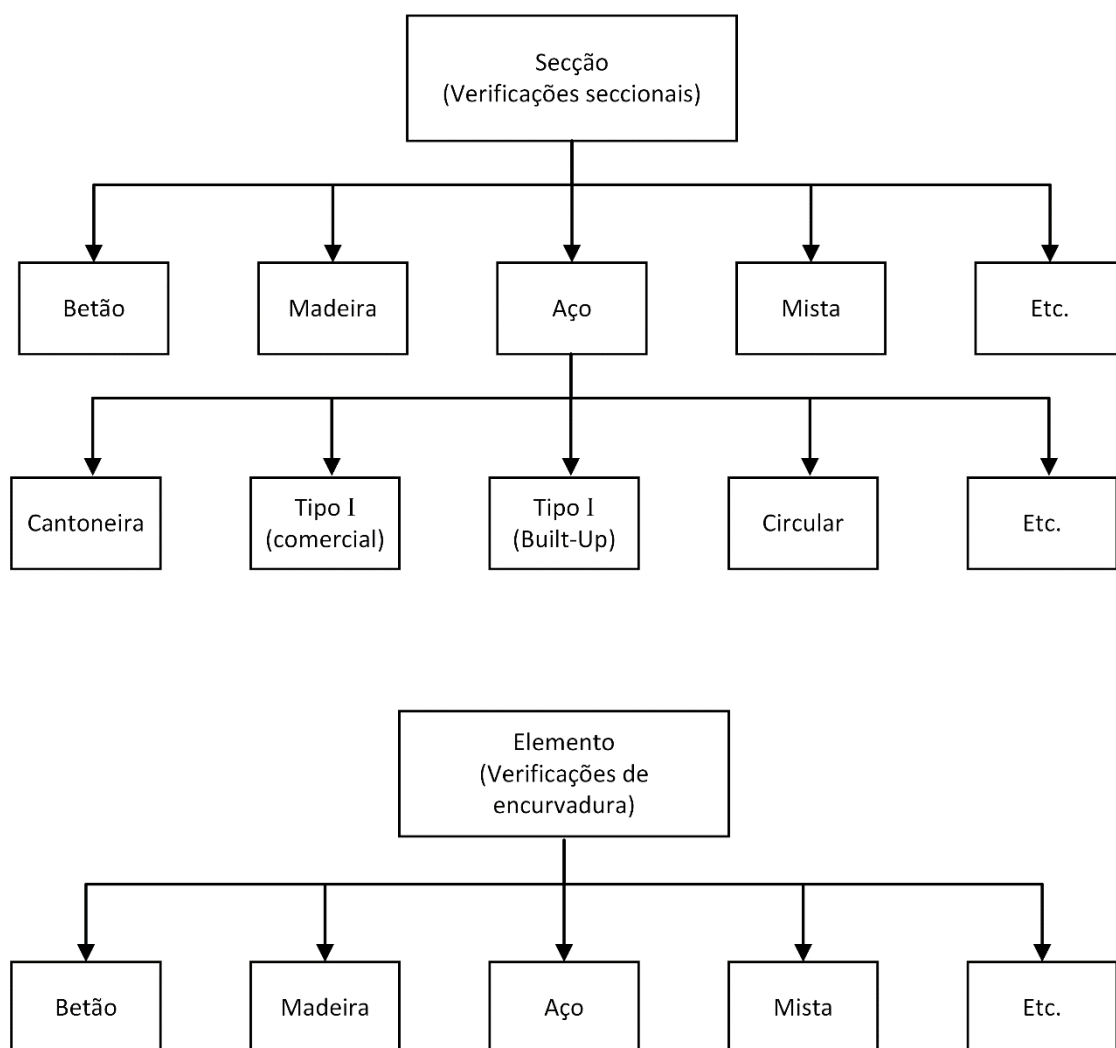


Fig. 3.8 - Esquema da arquitetura idealizada para o módulo

3.4.3.1. Secções metálicas

Neste módulo, os objetos do tipo secção estão definidos de modo a que, perante a introdução de uma secção tabelada e mediante um conjunto de esforços, classifiquem a secção e obtenham os valores dos esforços resistentes através das equações (2.3) a (2.6) considerando, sempre que aplicável, as interações do esforço axial e do esforço transversal com o momento resistente. Neste subcapítulo serão explicados os procedimentos que, embora sigam o EC3, não o fazem de forma expedita.

- Classificação

A classificação de secções sugerida no EC3 foi pensada de modo a classificar secções em estado limite (situação de dimensionamento clássico), contudo o processo de experimentação inerente a um algoritmo de otimização implica o teste de secções cujos esforços poderão estar altamente sub ou sobre dimensionados. Assim, para verificar as secções, optou-se por as classificar segundo a interseção do gradiente das forças aplicadas com a curva de interação, conforme é exemplificado na Fig. 3.9.

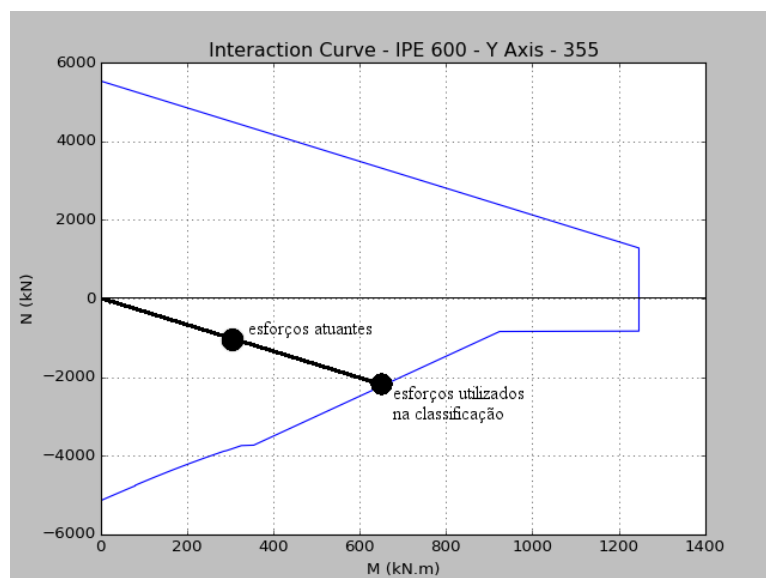


Fig. 3.9 - Exemplo de aplicação da classificação a uma viga IPE600 com $N_{ed} = -1000\text{kN}$ e $M_{ed} = 300\text{kNm}$

- Verificações de segurança

Considerando que estas podem ser aplicadas a um caso genérico, a verificação de uma secção resume-se a duas funções: verificação de momentos com interação de esforço axial e verificação de momentos com interação esforço transversal, caso exista interação, de acordo com as expressões definidas no EC3 para secções em flexão composta.

3.4.3.2. Elementos metálicos

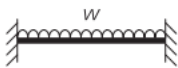





Os objetos do tipo elemento são constituídos por um comprimento e um objeto do tipo secção. Os elementos metálicos estão pensados de modo a correr apenas uma função, verificação dos elementos à encurvadura de acordo com o EC3, que se traduz pela verificação da equação (2.11), e que utiliza o método Austro-Alemão previsto no Anexo B do EC3. Para a execução desta função deverão ser fornecidos os valores de C_1 , C_{ms} , N_{ed} , M_{ed} e l_{eff} . Onde C_{ms} é o valor de $C_{mx}=C_{my}=C_{mLT}$, assumindo que esta igualdade se verifica.

- Verificação dos elementos à encurvadura

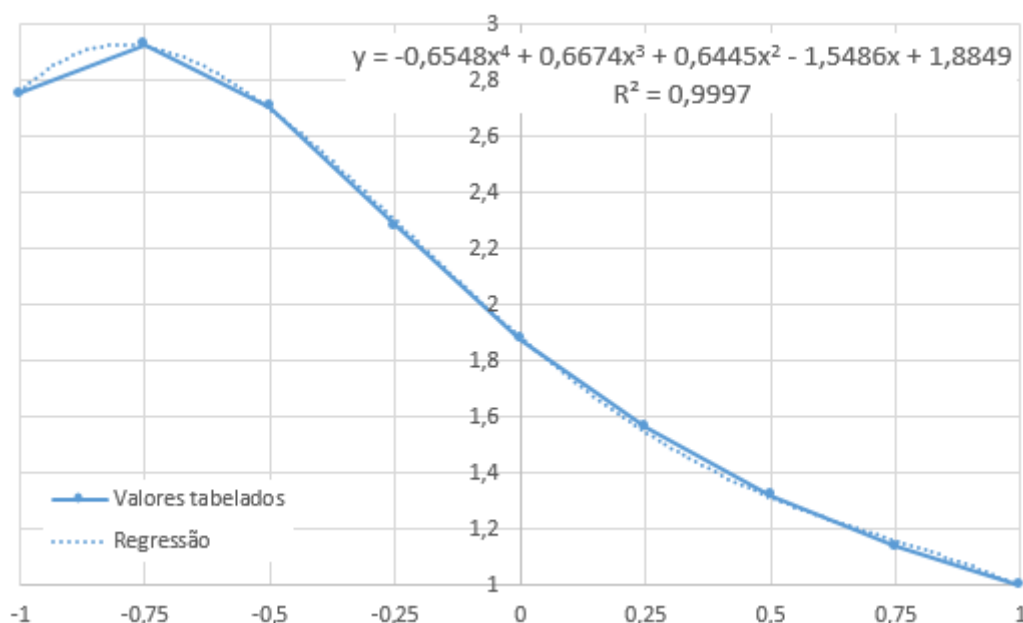
Na verificação da equação supracitada, o cálculo do valor do momento crítico, M_{cr} , foi implementado utilizando a expressão (3.1), onde para C_1 se utilizou uma expressão obtida por regressão dos valores apresentados na Tabela 6.10 do *Designer's guide to EC3* [17] para os pilares, e usando a Tabela 4 para as vigas, da qual se utiliza o valor mais conservativo se existirem cargas com distribuições diferentes no mesmo elemento.

$$M_{cr} = C_1 * \frac{\pi^2 * E * I_z}{L_{eff}^2} * \sqrt{\frac{I_w}{I_z} + \frac{L_{eff}^2 * G * I_z}{\pi * E * I_z}} \quad (3.1)$$

Tabela 4 - Excerto da Tabela 6.11 do *Designer's guide to EC3* [17]

Tipo de carregamento	Diagrama de momentos fletores	Valor de C_1
		1.285
		1.565
		1.046

Efetuada a regressão dos valores da tabela do *Designer's guide* [17], a expressão que devolveu valores mais satisfatórios da relação R-squared, sem comprometer a eficiência do algoritmo, foi uma regressão polinomial de 4º grau, tal como ilustrado na Fig. 3.10. Assim, a aproximação usada foi (3.2), à qual corresponde um coeficiente de correlação de 0.9997.

Fig. 3.10 - Regressão linear de C_1

$$C_1 = -0.6548\psi^4 + 0.6674\psi^3 + 0.6445\psi^2 - 1.5486\psi + 1.8849 \quad (3.2)$$

Os parâmetros C_{my} , C_{mz} e C_{mLT} são calculados de acordo com o Anexo B do Eurocódigo 3. Estes são considerados iguais e obtidos por consulta do quadro B.3 do EC3 e, tendo em conta que este é o caso mais conservativo e que o impacto nos resultados é muito reduzido, são sempre adotados os valores aplicáveis a cargas uniformes.

3.5. ALGORITMOS DE OTIMIZAÇÃO

Numa fase inicial do programa optou-se por implementar dois algoritmos diferentes de otimização de estruturas. O primeiro é um algoritmo básico que, mediante um conjunto de dados, realiza um dimensionamento linear, considerando o coeficiente de comportamento indicado no EC8, e devolve uma estrutura que respeita os regulamentos, em pouco segundos, mas muito pouco otimizada. O segundo é um algoritmo que realiza uma pesquisa harmónica das soluções possíveis e devolve uma estrutura otimizada, permitindo a utilização do método de dimensionamento IFBD, referido na Secção 2.3.6. Contudo, este algoritmo demora algumas horas a encontrar uma solução.

3.5.1. ALGORITMO BÁSICO

O algoritmo básico procura oferecer uma solução célere para o problema. Trata-se de um método mais limitado em termos de amplitude de opções e do resultado final, contudo fornece soluções que cumprem com todos os requisitos regulamentares. O processo começa com a atribuição da secção mais baixa possível para cada elemento, de seguida são realizadas iterações onde a secção de um determinado elemento é aumentada sempre que este não cumpra uma das condições estabelecidas, até ser obtida uma estrutura que verifique todas as condições impostas pelos Eurocódigos. O processo é ilustrado, através de um fluxograma, na Fig. 3.11.

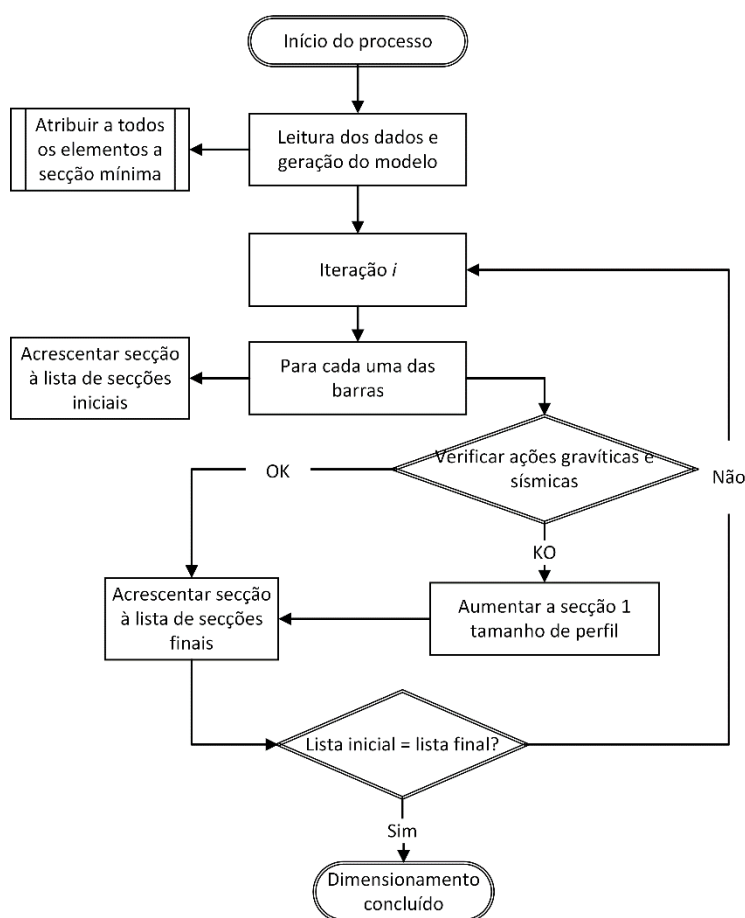


Fig. 3.11 - Fluxograma do processo associado ao método básico

3.5.2. PESQUISA HARMÓNICA

Com o objetivo de obter uma solução o mais otimizada possível em termos de peso de aço foi implementado um algoritmo de pesquisa harmónica (Harmony Search [18]). Existem várias publicações que provam a aplicabilidade de algoritmos harmónicos a estruturas metálicas, assim como a qualidade do seu desempenho, como é o caso das publicações de Saka [14] e Degertekin [9]. No entanto, não foi observado um padrão nos parâmetros de configuração do algoritmo nem a aplicação deste a problemas com o número de variáveis que se procurava implementar.

3.5.2.1. Formulação do problema

Um problema de otimização de um pórtico metálico é um problema de variáveis discretas, pois cada variável (barra) tem um conjunto definido de soluções possíveis (conjunto de perfis catalogados), cujo objetivo é a minimização do peso. Assim, a função objetivo traduz-se da seguinte forma:

$$\text{Minimize } W(x) = \sum_{k=1}^{N_{\text{elementos}}} A_k * L_k * \rho \quad (3.3)$$

A fórmula anterior permite calcular o somatório do produto do volume de todas as barras ($A_k * L_k$) com o peso volúmico do aço (ρ), sendo este último indicado pelo utilizador na fase de introdução de dados. Contudo, este tipo de algoritmos foi pensado para resolver funções sem restrições. Uma vez que os problemas de engenharia civil estão enquadrados no âmbito de um conjunto de regras, é necessário converte-los a um problema sem restrições equivalente. Para realizar esta conversão foi implementado um conjunto de critérios de penalização da função objetivo. Normalmente, estes critérios traduzem-se pelo aumento do peso da estrutura em função do número de restrições em incumprimento. Neste trabalho foi implementada uma variação da função de penalização adaptada das propostas por Pezeshk, *et al.* [13] e Camp, *et al.* [8], que é representada na equação (3.4):

$$\varphi(x) = W(x) * [1 + C]^\epsilon \quad (3.4)$$

Onde C é o resultado da função de violação de critérios, e ϵ é o expoente da função de penalizações. A função de violação de critérios é traduzida pela seguinte equação:

$$C = 5 * \sum_{i=1}^6 \frac{C_i}{\sum_{i=1}^6 C_i} * C_i = \frac{5}{\sum_{i=1}^6 C_i} * \sum_{i=1}^6 C_i^2 \quad (3.5)$$

Os coeficiente C_i podem assumir valores entre 0 e 1, consoante a razão entre o número de parâmetros respeitados e o número total de parâmetros, conforme explicado na Tabela 5, em que N_i significa “número de parâmetros que verificam i ”.

Tabela 5 - Cálculo dos critérios de penalização

Verificações sísmicas	
Limitação de danos	$C_1 = \frac{N_{pisos\ KO}}{N_{pisos}}$
Verificação de segurança dos elementos estruturais	$C_2 = \frac{N_{barras\ KO}}{N_{barras}}$
Verificação da condição de ductilidade local	$C_3 = \frac{N_{nós\ KO}}{N_{nós}}$
Coeficiente de sensibilidade – efeito não linear geométrico	$C_6 = \frac{N_{pisos\ KO}}{N_{pisos}}$
Verificações gravíticas	
Estado limite último e de segurança	$C_4 = \frac{N_{barras\ KO\ (ELU)} + N_{barras\ KO\ (ELS)}}{2 * N_{barras}}$
Condição geométrica	
Redução em altura	$C_5 = \frac{N_{transições\ que\ verificam}}{N_{transições}}$

A utilização dos parâmetros e critérios supracitados devolveu resultados bastante satisfatórios para estruturas até 8 pisos, o qual se considera altura limite para aplicação do método das forças laterais equivalentes. De seguida, é apresentado de forma conceptual a aplicação do algoritmo a uma estrutura de 5 pisos com 3 vãos (Fig. 3.12).

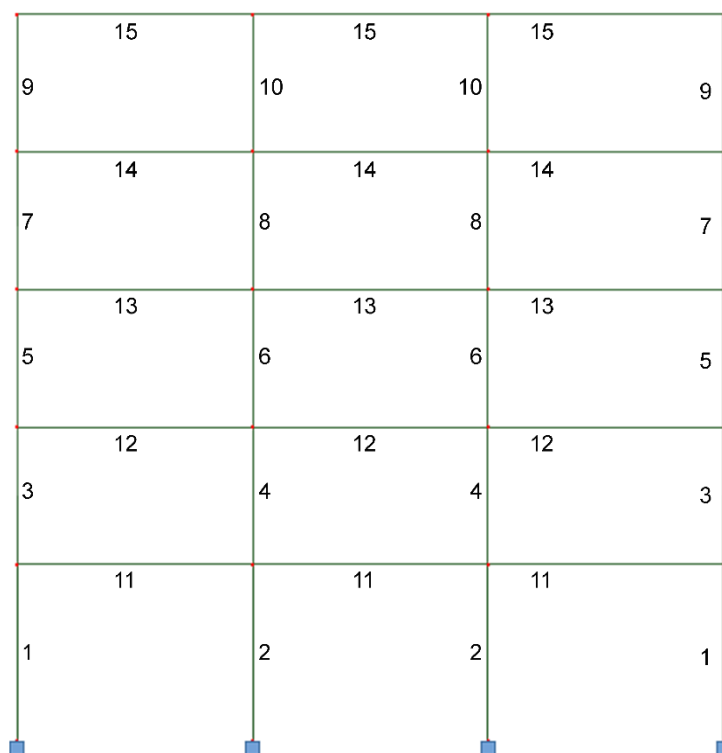


Fig. 3.12 - Estrutura de 3 vão com 5 pisos

Assumindo que os grupos de pilares podem tomar o valor de qualquer um dos perfis da série HEB (24 perfis) e os grupos de vigas qualquer um dos perfis da série IPE com pelo menos 100mm de altura (17 perfis) existem 90 triliões (10^{18}) de possibilidades de solução. Considerando que cada análise completa demora 2 a 3 segundos, dependendo do tipo de CPU, torna impraticável a obtenção do mínimo absoluto do problema. Contudo, um algoritmo evolutivo consegue obter uma solução extremamente próxima da ótima com um número razoável de iterações. Para este exemplo, que será posteriormente desenvolvido nos estudos de caso, o algoritmo de pesquisa harmônica conseguiu obter uma estrutura de 10.57 toneladas ($q = 4$) com apenas 7500 iterações, tendo sido a duração total do processo de 5 horas e 7 minutos.

3.5.2.2. OBTENÇÃO DOS PARÂMETROS DE CARACTERIZAÇÃO DA PESQUISA HARMÔNICA

Assim, foram realizados vários testes de sensibilidade dos parâmetros, incluindo o impacto da utilização de parâmetros adaptativos. Na Tabela 6 são apresentados os vários resultados obtidos para algumas das hipóteses testadas, os testes foram realizados para uma estrutura de 3 vãos com 5 pisos, utilizando um critério de simetria para vigas e pilares, resultando num total de 20 variáveis (10 pilares e 10 vigas). Estes testes foram realizados numa versão primária do programa, pelo que os valores finais de pesos poderão ser ligeiramente diferentes dos obtidos para a mesma estrutura na versão atual.

Ao aplicar o método proposto por Kumar, *et al.* [11] verificou-se que, embora os resultados fossem mais satisfatórios que os obtidos com parâmetros estáticos, o melhor resultado era alcançado muito próximo da última iteração. Este método aplica um decréscimo exponencial do *Pitch Adjustment Rate* (PAR) e um crescimento linear do *Harmony Memory Consideration Rate* (HMCR), conforme ilustrado na Fig. 3.13. Deste modo mostrou-se oportuno testar as hipóteses de considerar um maior número de iterações ou um patamar, após a conclusão do método de Kumar, *et al.*, no qual se mantem os valores mínimos e máximos de PAR e HMCR até se verificar um determinado número de iterações sem melhoria da melhor solução. Considerar um patamar até ocorrerem, sem alteração da melhor solução, um total de 10% do número máximo de improvisações mostrou resultados muito satisfatórios para um número de iterações relativamente reduzido, conforme pode ser verificado na Tabela 6.

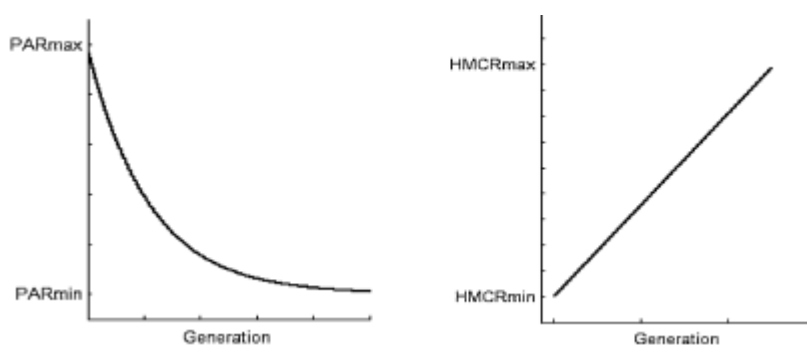


Fig. 3.13 - Evolução dos parâmetros proposta por Kumar, *et al.* [11]

Tabela 6 - Resultados dos testes de sensibilidade dos parâmetros do HS

Método	HMS	HMCR	HMCR (min)	HMCR (max)	PAR	PAR (min)	PAR (max)	BW	BW (min)	BW (max)	Iteração da melhor solução	Total de iterações	Peso total (kg)
Proposto	20	-	0,7	0,99	0	0,01	0,99	0	1	4	5862	6362	12915,6
Kumar, et al. [11]	10	-	0,7	0,99	-	0,01	0,99	-	1	4	4847	5000	13133,0
Proposto	10	-	0,7	0,99	-	0,01	0,99	-	1	4	5461	5556	13159,9
Kumar, et al. [11]	10	-	0,7	0,99	-	0,01	0,99	-	1	17*	4859	5000	13975,1
Original	10	0,95	-	-	0,3	-	-	4	-	-	4004	5000	13981,0
Kumar, et al. [11]	50	-	0,7	0,99	-	0,01	0,99	-	1	4	4990	5000	14096,9
Kumar, et al. [11]	100	-	0,7	0,99	-	0,01	0,99	-	1	4	4370	5000	14796,1
Kumar, et al. [11]	60	-	0,85	0,99	-	0,1	0,5	-	1	4	4965	5000	14811,2
Original	100	0,8	-	-	0,4	-	-	3	-	-	2064	5000	20147,7

*dimensão máxima da menor lista de possíveis perfis

Assim, a metodologia de pesquisa harmónica implementada neste trabalho utiliza parâmetros adaptativos obtidos pelas equações (2.28) a (2.30), até um limite de iterações definido pela equação (3.6), nos quais é aplicado o método proposto por Kumar, *et al.*, conforme representado na Fig. 3.14. Os intervalos entre os quais os valores poderiam variar também foram objeto de estudo, concluindo-se que para os casos de PAR e HMCR os valores utilizados por Kumar, *et al.* devolviam resultados satisfatórios, estando estes no intervalo [0.01,0.99] e [0.7,0.99], respetivamente. Para os valores de BW concluiu-se que utilizar um intervalo de ajustes entre 1 e 20% da dimensão média das listas de perfis devolvia resultados satisfatórios para um menor número de iterações.

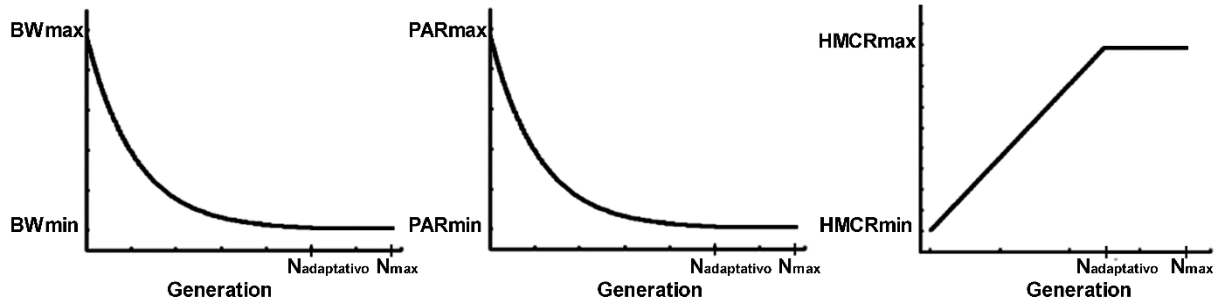


Fig. 3.14 – Representação gráfica dos parâmetros utilizados no presente trabalho

Com o objetivo de encontrar um equilíbrio entre o tempo de análise e a qualidade dos resultados, foram definidos dois critérios de paragem: o primeiro é um número máximo de improvisações obtido em função do número de variáveis, de acordo com a equação (3.7), o segundo é um critério de estabilidade que interrompe o processo caso decorram 20% do número máximo de improvisações, equação (3.7), sem alteração da melhor solução.

Encontrar um equilíbrio entre o número máximo de improvisações e a dimensão da memória harmónica (HMS) foi um processo bastante complexo, pois o algoritmo é altamente sensível a estes dois parâmetros. Naturalmente, quanto maior o número de iterações, mais leve será a solução encontrada, porém maior será o tempo despendido no processo. Os vários testes realizados revelaram valores satisfatórios quando estes parâmetros eram calculados do seguinte modo:

$$N_{\text{improvisações a aplicar}} = 0.0305 * e^{0.1556 * N_{\text{variáveis}}} * N_{\text{variáveis}} * 1000 \quad (3.6)$$

parametros adaptativos

$$N_{\text{máximo de improvisações}} = N_{\text{variáveis}} * 1000 \quad (3.7)$$

$$HMS = \text{inteiro}(N_{\text{variáveis}} * 2.5) \quad (3.8)$$

4

DOCUMENTAÇÃO DA FERRAMENTA DE CÁLCULO

4.1. INTRODUÇÃO

Este capítulo, escrito com o objetivo de proporcionar uma compreensão do programa e suas funções, procura facilitar o desenvolvimento de trabalhos futuros. A linguagem utilizada é mais técnica e específica de Python. Serão também apresentados alguns exemplos em pseudo-código para demonstrar o funcionamento de algumas funções.

O programa está estruturado com uma classe "mãe", *Model*, que herda atributos de 4 classes secundárias, *Interface*, *Gravitic*, *Dynamic* e *Optimization*. A classe *Model* agrega toda a informação relativa ao modelo da estrutura que é criado e trabalhado. A classe *Interface* é constituída pelas funções de interação do programa com o exterior. As classes *Gravitic* e *Dynamic* são constituídas pelas funções responsáveis pelas análises gravíticas e dinâmicas, respetivamente. Por último, a classe *Optimization* implementa o algoritmo de otimização. Na Fig. 4.1 é apresentado um esquema global da arquitetura do programa.

4.2. CLASSE MODEL

O modelo é a classe central do programa. O objeto da classe *Model* é responsável por guardar os vários atributos do modelo, gerir a informação e executar as várias funções definidas nas classes das quais este é herdeiro. Assim, o processo deverá ser inicializado criando um objeto da classe *Model*, através do seguinte comando:

Excerto de código 1: Criação do modelo

```
1.      Model(filename, beam_sec_list, col_sec_list, log)
```

Para a criação do modelo de informação deverá ser fornecida a localização do ficheiro de Excel com as informações da estrutura a dimensionar (*filename*), para o qual é esperado um valor do tipo *string*. Deverão, também, ser fornecidas as listas de possíveis perfis a iterar no dimensionamento, tanto para vigas como para pilares, por ordem de preferência (no caso do algoritmo básico); assim como ser pedido um registo das operações realizadas durante a execução do programa, definindo *log = True*.

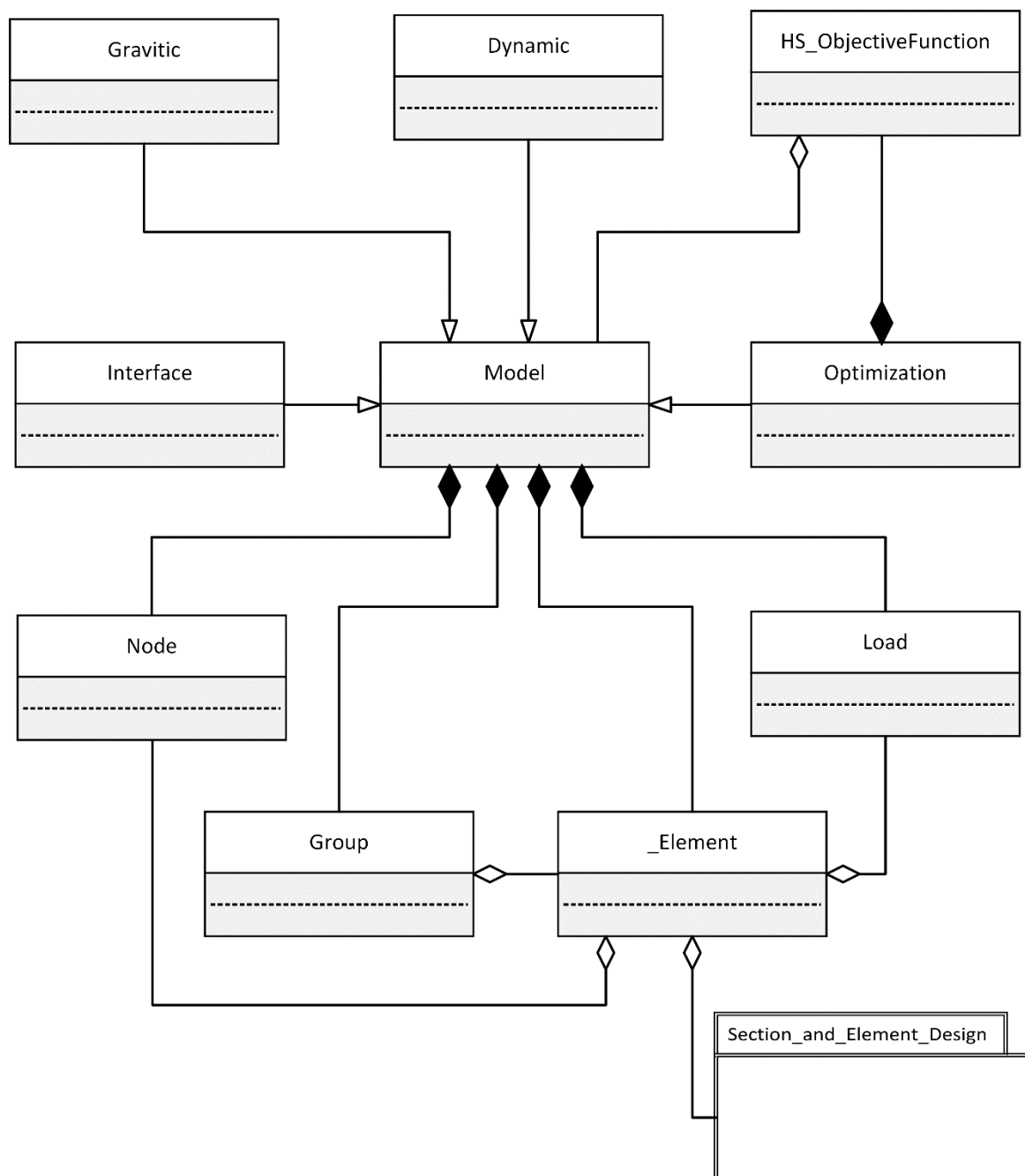


Fig. 4.1 - Esquema global de funcionamento do programa

4.2.1. MÉTODO __INIT__

A criação de um objeto da presente classe atribui-lhe automaticamente os dados fornecidos. Processando, de seguida, o ficheiro de Excel e criando o modelo estrutural, funções que serão descritas na Secção 4.3. Por fim, este método encaminha para o método que gere o processo definido pelo utilizador: *basic_design* ou *harmony_search_opt*.

4.2.2. MÉTODO _DESIGN_CONDITION

Este método é responsável pela implementação das restrições de dimensionamento na direção horizontal. À data da realização deste documento apenas está implementada uma condição de simetria segundo o eixo central. Contudo, o método está preparado para a implementação de novas condições, como pode ser observado no excerto exemplificativo subsequente.

Excerto de código 2: Aplicação da condição de simetria

```

1.     if self.d_condition == 'Symmetrical':
2.         pair = self.ele_range_storey[b_c][element/100-100*(b_c+1)]
           [-self.ele_range_storey[b_c][element/100-100*(b_c+1)].index(element)-1]
3.         if b_c == 0:
4.             if self.beams[element].section != self.beams[pair].section:
5.                 self.beams[pair].section = self.beams[element].section
6.                 self.beams[pair].refresh_section_props()
7.         elif b_c == 1:
8.             if self.columns[element].section != self.columns[pair].section:
9.                 self.columns[pair].section = self.columns[element].section
10.            self.columns[pair].refresh_section_props()

```

Note-se que este método apenas é utilizado pelo algoritmo básico, com o objetivo de forçar a alteração da secção de barras dependentes entre si.

4.2.3. MÉTODO _REDUCE_IN_HEIGHT

À semelhança do método anterior, o método *reduce_in_height* também tem implementado apenas uma possibilidade de condição, que é a redução de área da secção em altura, e está preparado, se necessário, para a implementação de outras condições. Contudo, no âmbito da implementação de algoritmos de verificação de segurança, este método também possui um quantificador da razão do número de elementos que respeitam a condição de segurança sobre o número total de elementos.

4.2.4. MÉTODO TOTAL_WEIGHT

Método utilizado para calcular o peso total da estrutura, sem penalizações, de acordo com a equação (3.3).

4.2.5. MÉTODO BASIC_DESIGN

O método *basic_design* implementa o processo descrito em 3.5.1. Trata-se de um dimensionamento linear da estrutura, embora pouco otimizado. Assim, é um método recursivo que realiza a análise gravítica e sísmica, aumentando as secções caso estas não sejam adequadas. De seguida verifica a existência de pisos recuados com o objetivo de analisar a estrutura para forças sísmica de sentido oposto, se esta não for simétrica. Por fim, verifica se as secções da estrutura foram alteradas desde o início da iteração. Caso se verifique, a função é executada novamente.

4.3. CLASSE INTERFACE

Uma vez que os métodos desta classe apenas são chamados pontualmente no programa, maioritariamente quando é realizado um *input* ou *output*, os métodos desta classe fazem uso de várias funções. Este facto deve-se ao objetivo de manter o código organizado e facilmente legível, uma vez que muitas das funções associadas à interface são muito extensas, tendo sido sempre ponderado o interesse entre usar o método ou a função.

4.3.1. LEITURA DE DADOS E CRIAÇÃO DO MODELO

Os dados do problema, introduzidos numa folha de Excel, são interpretados pelo método *parsing*. Este método utiliza três funções para obter os dados gerais (geométrico, materiais, etc.) e atribui ao modelo o dicionário de cargas e dicionário de grupos de vigas/pilares. Note-se que os componentes do modelo (nós, elementos, cargas e grupos) são armazenados em dicionários Python, nos quais os objetos são devidamente identificados com o sistema de catalogação explicado no Capítulo 3.

De seguida é executado o método *generate*. Este método utiliza os dados obtidos pela primeira função do *parsing* para gerar os nós e elementos do modelo, com recurso também a funções. Deste método resulta a atribuição ao modelo dos dicionários de nós, vigas e pilares, devidamente catalogados. Seguidamente é executada a função que elimina dos respetivos dicionários os objetos inexistentes devido aos pisos recuados. Por fim é executado o método *imperfections_for_global_analysis* com o propósito de aplicar as imperfeições iniciais às coordenadas dos nós. Este último método, embora o cálculo dos valores de imperfeição por barra se resuma ao excerto apresentado, é um pouco extenso pois verifica se as imperfeições aplicadas anteriormente são válidas (em função do número de pilares cujo esforço axial é superior a 50% da média do piso) e, por fim, soma as variações de posição em altura para obter os valores finais de variação.

Excerto de código 3: Aplicação das imperfeições globais

```
1.     for i in self.columns:
2.         alpha = min(max(2./self.columns[i].length**0.5,2./3),1)
3.         teta = 1./200*alpha*(0.5*(1+1./m))**0.5
4.         delta = teta*self.columns[i].length
5.         self.nodes[self.columns[i].noj].delta_x = delta
        [...]
58.    for i in sorted(self.nodes):
59.        if i/100 == 300:           #fixed nodes
60.            pass
61.        else:
62.            self.nodes[i].delta_x += self.nodes[i-100].delta_x
```

No que concerne à leitura de dados existe, ainda, uma opção facultativa no método de *output* da estrutura que permite ao utilizador confirmar a geometria e as ações na sua estrutura, *graphic_model(phase='Input')*. Assim, são representadas duas estruturas, a da esquerda com as cargas nos elementos e a da direita com as cargas nos nós. Um exemplo deste método opcional é apresentado na Fig. 4.2.

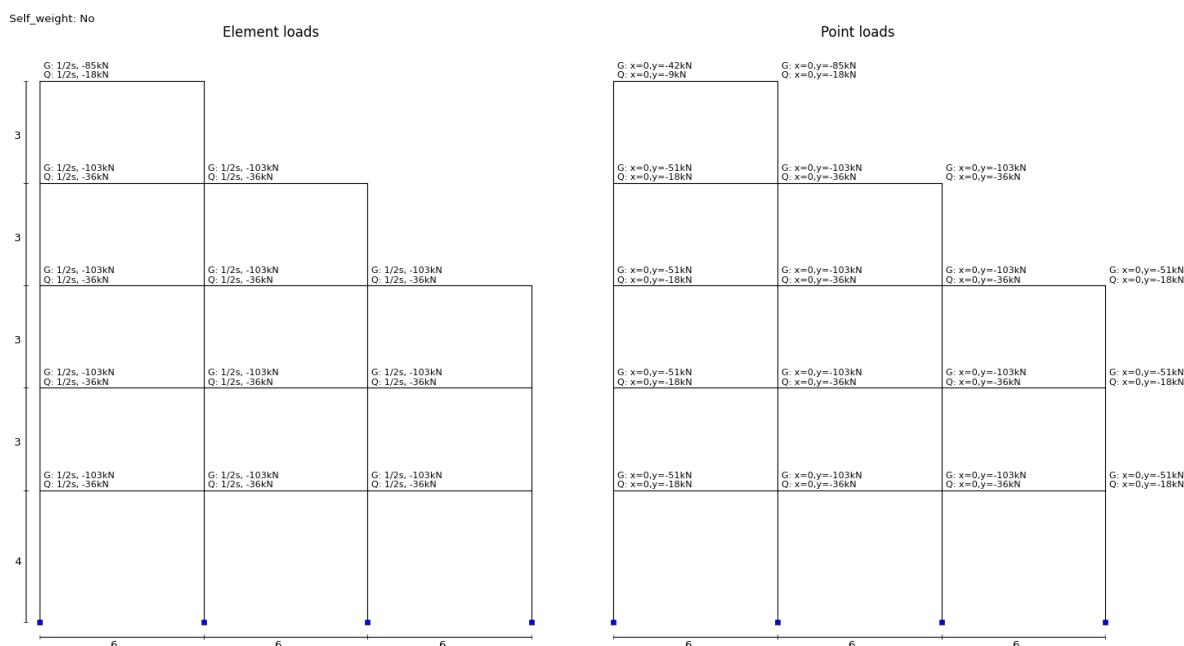


Fig. 4.2 - Representação da estrutura introduzida pelo utilizador

4.3.2. EXPORTAÇÃO DE RESULTADOS

Atualmente, na exportação de resultados utiliza-se o método *graphic_model(phase='Output')* que gera uma imagem contendo a estrutura, as secções atribuídas e algumas propriedades dinâmicas desta. Esta informação é acompanhada dos dados na consola sobre o peso, tempo de análise e resultados dos coeficientes de penalização, conforme exemplificado na Fig. 4.3. São também exportados para ficheiros do texto os textos imprimidos para a consola, através do método *record_log*, e das barras e propriedades dinâmicas, através do método *result_export*. Contudo, já está a ser pensada a atualização deste último método para produzir uma memória de cálculo.

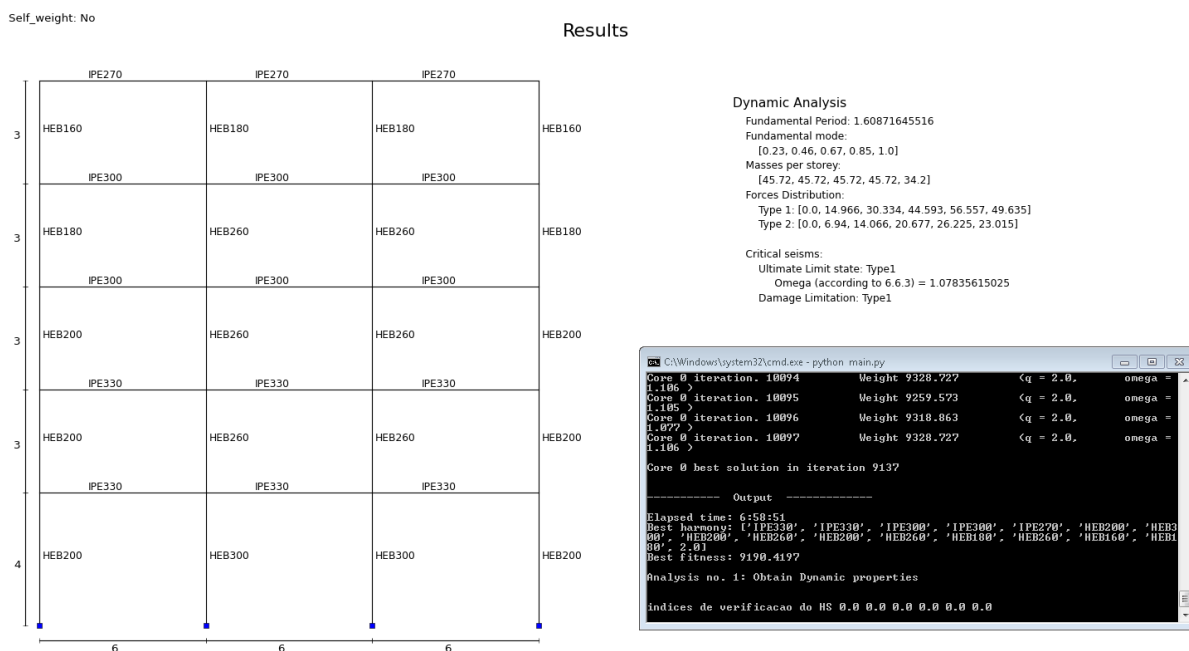


Fig. 4.3 – Exemplo de Output gráfico de uma pesquisa harmónica

O método *graphic_model*, utilizado tanto para representação dos dados inseridos como para exportação de resultados obtidos, executa a função *plot_struct*, a qual se apoia na biblioteca Python *matplotlib* para desenhar a estrutura e produzir textos que lhe estão associados.

4.3.3. INTERFACE COM O OPENSEES

A interface com o OpenSees é realizada com recurso a um motor de *templates* de Python, o Jinja2 [19]. Este módulo é baseado na *web application framework* Django [20], tendo assim uma linguagem muito semelhante à de Python, o que torna a sua aprendizagem relativamente fácil. Esta geração dos ficheiros de análise é realizada por várias funções de estrutura semelhante, implementadas no ficheiro *template_design.py*, que recebem como parâmetros os dicionários de nós, vigas, pilares, cargas e informação sobre o caso de carga e respetivos coeficientes associados. De seguida, consoante o objetivo da análise (gravítica, sísmica ou dinâmica), a função utiliza um *template* previamente definido para gerar os ficheiros do modelo de análise do OpenSees. Note-se que existe uma função específica para análise dinâmica que se destina a obter as propriedades dinâmicas da estrutura.

Excerto de código 4: Exemplo do *template* usado para gerar a geometria da estrutura no OpenSees

```

1.      #nodes $tag $Node,i $Node,j
2.      {% for i in nodes -%}
3.      node {{ nodes[i].tag }} {{ nodes[i].coordx_os() }}
   {{ nodes[i].coordy }} -mass 1e-9 {{ nodes[i].mass }} 0.0
4.      {% endfor %}
5.
6.      #fix $tag $dx $dy &dx
7.      {% for i in nodes -%}
8.      {% if nodes[i].tag/100 < 301 -%}
9.      fix {{i}} 1 1 1
10.     {% endif -%}
11.     {% endfor %}
12.
13.     set Es {{ mats[0] }}; # Elastic Modulus assigned to variable Es (kPa)
14.     geomTransf PDelta 1;
15.
16.     #element nonlinearBeamColumn $tag $Node,i $Node,j $numIntgrPts
   $secTag $transfTag
17.     {% for i in beam_list -%}
18.     element elasticBeamColumn {{ beams[i].tag }} {{ beams[i].noi }}
   {{ beams[i].noj }} {{ beams[i].area }} $Es {{ beams[i].Iy }} 1 -mass
   {{ beams[i].massDens }}
19.     {% endfor -%}
20.     {% for i in colu_list -%}
21.     element elasticBeamColumn {{ columns[i].tag }} {{ columns[i].noi }}
   {{ columns[i].noj }} {{ columns[i].area }} $Es {{ columns[i].Iy }} 1
22.     {% endfor %}

```

Por fim, é executada a função **run_OpenSees** que executa o OpenSees tendo como *input* o ficheiro gerado na função anterior. É recomendado ter o OpenSees na *path* do Windows para evitar erros na sua execução.

4.4. CLASSE GRAVITIC

Os métodos desta classe realizam a gestão das análises associada às combinações 1 a 4 da Tabela 3, ou seja, com ações base sobrecarga ou vento. Executam, assim, as verificações implementadas nos objetos do tipo elemento e tratam os resultados avaliando a estrutura relativamente às verificações de segurança do EC3.

4.4.1. MÉTODO GRAVITIC_VERIFICATION

Método utilizado para avaliar o coeficiente de penalização associado ao processo de otimização. Começa por definir duas variáveis, **number_of_verifications** e **number_of_KOs**, iguais a zero e corre um ciclo no qual verifica as combinações 1 a 4, somando 1 às variáveis referidas anteriormente sempre que seja realizada uma verificação e sempre que uma verificação seja negativa, respetivamente. Realiza, para as combinações 1 e 3, verificações dos deslocamentos a meio vão das vigas e, para as combinações 2 e 4, efetuando a verificação da resistência das secções e da encurvadura dos elementos. Este método devolve o rácio **number_of_KOs/number_of_verifications**.

4.4.2. MÉTODO GRAVITIC_ANALYSIS_AND_OPTIMIZATION

Método utilizado pelo algoritmo básico com o objetivo de verificar as combinações 1 a 4 e incrementar os perfis sempre que estes não verifiquem as condições impostas pelo EC3. As combinações são executadas pela seguinte ordem: 2 – 4 – 1 – 3, já que, para a maioria dos casos, esta será a ordem da mais condicionante para a menos condicionante. Para cada combinação é executado um ciclo *while* que visa garantir que todas as secções verificam os critérios do EC3. A função **design_check_and_optimize** é responsável pela verificação e aumento das secções para as combinações 2 e 4, enquanto a função **mid_span_disp_check** é responsável pela verificação de deformações das vigas e aumento das suas secções, caso necessário.

4.5. CLASSE DYNAMIC

Os métodos desta classe controlam o processo associado à obtenção das propriedades dinâmicas da estrutura e à sua análise sísmica.

4.5.1. INTRODUÇÃO DE DADOS E OBTENÇÃO DAS PROPRIEDADES SÍSMICAS

A leitura das folhas de Excel correspondentes à análise sísmica é realizada por um método desta classe, **dyn_input_parse**, que cria 2 dicionários com as características da ação do Tipo 1 e do Tipo 2, calcula os limites do coeficiente de comportamento em função da classe de ductilidade, calcula as massas por piso, caso estas não sejam definidas manualmente, e devolve os resultados. De seguida, o método **get_dyn_props** gera e executa um modelo OpenSees e obtém as propriedades dinâmicas da estrutura, atribuindo ao modelo o seu período fundamental, os *eigenvalues* e as massas por piso. Caso a forma de obtenção do coeficiente de comportamento, *q_type*, seja definida pelo método 'IFBD' este executa o

método *ifbd_apply* que confronta os esforços da estrutura com as resistências dos elementos e obtém o coeficiente de comportamento. Por fim, o método *lat_force_dist* distribui as forças sísmicas horizontais em altura, conforme definido pela equação (2.14).

4.5.2. MÉTODO SEISMIC_ANALYSIS_OS

Este método é responsável pela criação e execução das análises em OpenSees que incluem ações dinâmicas (combinações 5 a 10). Como resultado, a geração dos ficheiros a ser utilizados na execução das análises é realizada em duas fases: na primeira, é gerado o ficheiro do modelo e o ficheiro com as ações estáticas; na segunda fase, é gerado o ficheiro com as ações dinâmicas. Estes poderão ser utilizados na mesma análise, por exemplo para a verificação de *Capacity Design* (combinação 5), ou em duas análises diferentes, como é o caso das verificações de vigas ao estado limite último (combinações 9 e 10).

Excerto de código 5: Execução da análise para verificação dos pilares ao ELU (combinação 6)

```

1.     generate_os_file_seismic(nodes=self.nodes,beams=self.beams,
    columns=self.columns,mats=self.dado[5],loads=self.model_loads,
    coefs=self.coefs,case=5,process = self.process)
2.     [...]
3.         elif phase == 'ULS' and omega != None:
4.             if self.type2['Fb'] > self.type1['Fb']:
5.                 c_type = 'Type2'
6.             else: c_type = 'Type1'
7.             self.to_output[5]['ULS'] = c_type
8.             gamma_ov = 1.25
9.             self.count += 1
            generate_os_file_seismic(seismic_loads=
self.dyn_load[c_type],case=5,mult_factor=direction*1.1*gamma_ov*omega,
process= self.process)
10.         run_OpenSees(sourcefile=self.process+'OpenS_mainS.tcl')
11.         result_parse(self.beams,self.columns,self.nodes,
self.model_loads,self.coefs,case=6,at='dynamic',process = self.process)

```

4.5.3. VERIFICAÇÕES DE SEGURANÇA

As verificações no âmbito do EC8 são realizadas através de 4 métodos que verificam as condições de *capacity design*, estado limite último, limitação de danos e cálculo dos coeficientes de sensibilidade para os vários andares. Os dois primeiros estão implementados na íntegra para os dois métodos de dimensionamento, enquanto os dois últimos apenas estão implementados para a pesquisa harmónica (alvo principal deste trabalho).

4.5.3.1. Método seismic_check_ULS

A verificação ao estado limite último, implementada no método *seismic_check_ULS*, é realizada em dois passos. O primeiro consiste numa verificação viga a viga, da qual resulta um parâmetro booleano

(*True* se verificar, *False* se não verificar) sobre a verificação e o valor de sobrerresistencia da viga; o segundo consiste numa verificação pilar a pilar, para a qual é realizada uma análise em OpenSees para a combinação 6 (sendo Ω o menor dos valores obtidos no primeiro passo), resultando um parâmetro booleano sobre a validade da secção e do elemento. À semelhança do método anterior, este aumenta as secções em falha para o método básico e devolve o rácio *number_of_KOs/number_of_verifications* para a pesquisa harmónica.

4.5.3.2. Método *capacity_design_check*

O método *capacity_design_check* executa uma verificação nó a nó, na qual faz um somatório dos momentos resistentes das vigas e pilares que lhe são adjacentes e calcula, no final, o rácio destes, devolvendo um parâmetro booleano sobre o resultado da verificação da condição de ductilidade local. Se esta não for verificada, no caso do algoritmo básico procede para a incremento dos pilares do nó; no caso da pesquisa harmónica soma uma unidade ao *number_of_KOs* com a finalidade de, no final da verificação, devolver o rácio *number_of_KOs/number_of_verifications*.

4.5.3.3. Método *seismic_check_SLS*

Com uma implementação bastante direta, este método verifica a equação (2.27) andar a andar e devolve o rácio *number_of_KOs/number_of_verifications* em função do número de andares cujo deslocamento relativo entre pisos seja acima do permitido pelo regulamento.

4.5.3.4. Método *interstorey_drift_sensitivity*

No método *interstorey_drift_sensitivity* optou-se por calcular o coeficiente de sensibilidade (θ) para uma análise linear, por forma a ir de encontro à filosofia utilizada pelo EC8 na parametrização destas equações. Assim, inicialmente é executada uma análise linear em OpenSees para a combinação 5. De seguida, é verificada a equação (2.23) piso a piso considerando um valor máximo para θ de 0.3, em conformidade com o ponto 4.4.2.2(4) do EC8. Por fim, é devolvido o rácio entre o número de pisos que não respeitam a equação e o total de pisos.

Excerto de código 6: Representação da obtenção do coeficiente de penalização associado ao método *interstorey_drift_sensitivity*

```

1.     generate_os_file_seismic([...])
2.     run_OpenSees('OpenS_mainS.tcl')
3.     drift = np.loadtxt('Results_OpenS_dynamic/displacement.out')
4.     #calculate thetas
5.     thetas = {}
6.     for i in sorted(self.ele_range_storey['columns']):
7.         h = self.columns[self.ele_range_storey['columns'][i][0]].length
8.         dr = (drift[i]-drift[i-1])
9.         for j in self.ele_range_storey[1][i]:
10.            p += abs(self.columns[j].internal_forces[9][0][0])
11.            v += abs(self.columns[j].internal_forces[10][0][1])
12.            thetas[i] = p*dr*self.q/(v*h)
13.

```

```

14.     for i in thetas:
15.         if thetas[i] > 0.3: number_of_KOs += 1
16.     return float(number_of_KOs)/len(thetas)

```

4.5.4. MÉTODO SEISMIC_VERIFICATION

O método *seismic_verification* é responsável pela gestão da análise sísmica no âmbito da pesquisa harmónica, bem como por executar as funções supracitadas na ordem correta, de forma a devolver à função objetivo os coeficientes C_1 , C_2 e C_3 (o método *interstorey_drift_sensitivity* – C6 - é chamado diretamente pela função objetivo). O esquema de funcionamento do método é apresentado no fluxograma seguinte.

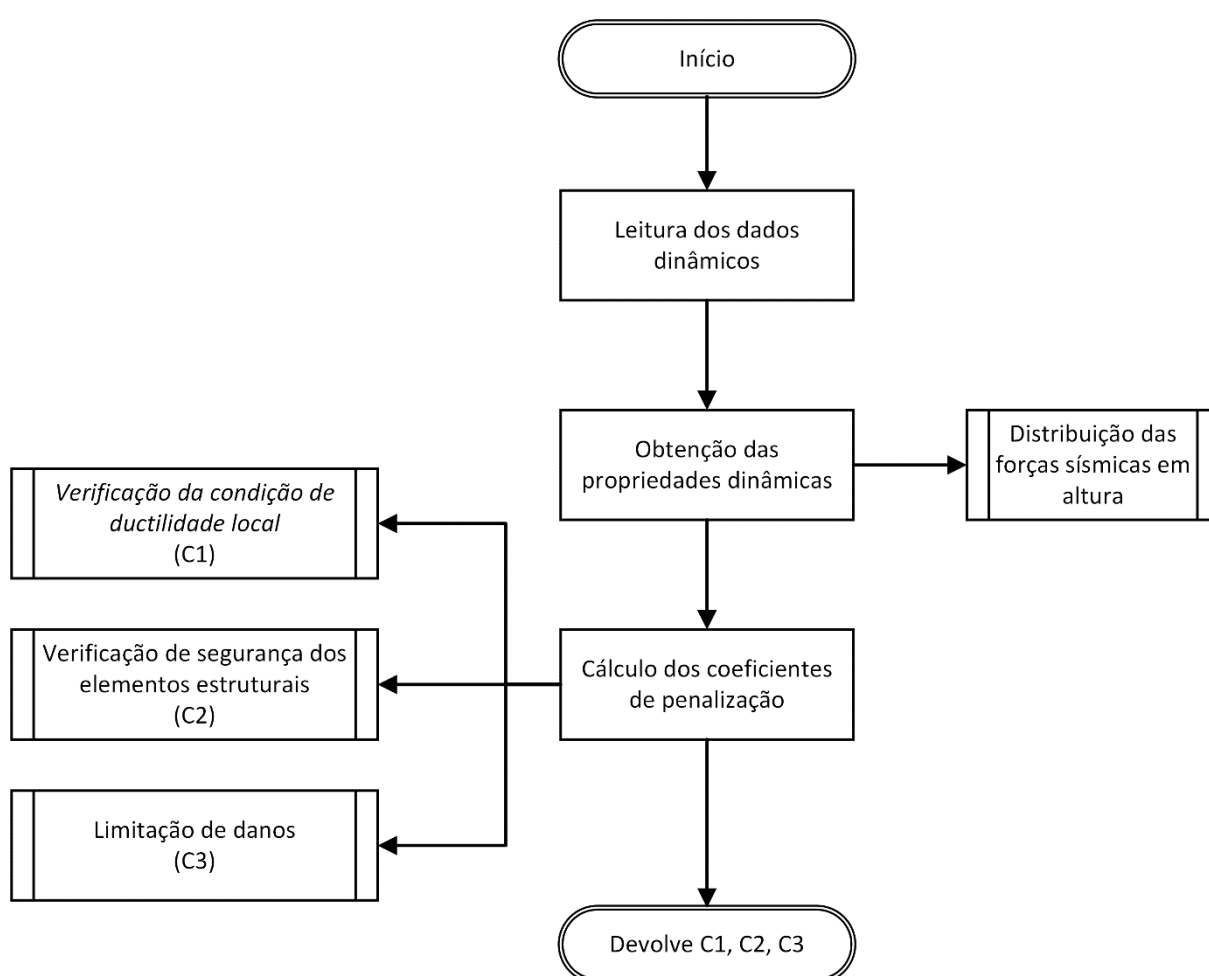


Fig. 4.4 - Fluxograma do método seismic_verification

4.5.5. MÉTODO SEISMIC_ANALYSIS_AND_OPTIMIZATION

À semelhança do método anterior, também este gere as verificações requeridas pelo EC8, contudo segue uma filosofia semelhante ao *gravitic_analysis_and_optimization*, aumentando as secções sempre que alguma verificação não seja cumprida. Este método apenas é utilizado pelo algoritmo básico.

4.6. CLASSE OPTIMIZATION

A classe *Optimization* visa a implementação e gestão de algoritmos de otimização. O algoritmo de otimização implementado, pesquisa harmónica (HS), utiliza, para realizar a análise, um objeto da classe *HS_ObjectiveFunction* que deverá conter toda a parametrização do algoritmo, desde o cálculo da função objetivo aos parâmetros de entrada.

4.6.1. MÉTODOS DA CLASSE OPTIMIZATION

A classe *Optimization* é constituída por três métodos que iniciam o algoritmo HS e interpretam os resultados obtidos. O método *starting_point* devolve, em função do tipo de estrutura e das restrições do problema, uma lista com o número de variáveis correspondentes a vigas e o número de variáveis correspondentes a pilares, respetivamente. Assim, o processo de otimização é iniciado com o método *harmony_search_opt* que executa o método *starting_point*, cria um objeto do tipo *HS_ObjectiveFunction* e inicia a análise. Note-se que um dos parâmetros de entrada na criação dos objetos da classe *HS_ObjectiveFunction* é uma cópia do próprio modelo. Isto permite um acesso mais eficiente aos vários métodos necessários à classificação das várias soluções e uma melhor compartimentalização dos modelos de informação no caso de serem utilizados vários *cores* na análise. Por fim, é executado o método *update_props* que utiliza a função objetivo como meio de fazer a atualização das secções, classificando e devolvendo os resultados.

4.6.2. MÉTODO HS_OBJECTIVEFUNCTION

Esta classe é adaptada da classe base fornecida pela biblioteca *pyHarmonySearch* [21], mantendo a estrutura original, com a adição de alguns métodos para atribuir as secções às barras em função da solução gerada pelo algoritmo.

Ao iniciar o objeto, é criada a lista de variáveis a iterar e são atribuídos os parâmetros que controlam o algoritmo. Conforme referido anteriormente, foram implementados parâmetros adaptativos. Contudo, foi criado um atributo que permite a sua anulação (esta poderá ter interesse para problemas com poucas variáveis). Os parâmetros de inicialização são explicados no excerto seguinte.

Excerto de código 7: Definição dos parâmetros da pesquisa harmónica

```

1.     self._discrete_values #lista de opções por variável
2.     #config:[[PerfisViga1],[PerfisViga2],...,[PerfisPilar1],[PerfisPilarN]]
3.
4.     #Parâmetros de input
5.     self._static = False   #parâmetros estáticos?
6.     self._maximize = False #maximizar ou minimizar?
7.     self._max_imp = 10000  #número mínimo de improvisações
8.     self._hms = 20         #Dimensão da memória harmónica
9.     self._hmcr = 0.9       #harmony memory considering rate
10.    self._par = 0.3         #pitch adjusting rate
11.    self._mpai = 4          #pitch adjustment index ou bandwidth
12.
13.    #Parameter adaptive
14.    self._hmcr_min = 0.7    #harmony memory considering rate mínimo

```

```
15.     self._hmc_r_max = 0.99    #harmony memory considering rate máximo
16.     self._par_min = 0.01     #pitch adjusting rate mínimo
17.     self._par_max = 0.99     #pitch adjusting rate máximo
18.     self._bw_min = 1         #pitch adjustment index ou bandwidth mínimo
19.     self._bw_max = 4         #pitch adjustment index ou bandwidth máximo
20.
21.     self._exp = 2            #expoente das penalizações
```

O método *get_fitness* é chamado pelo algoritmo de pesquisa harmónica com o objetivo de avaliar a solução gerada. Assim, este método recebe o vetor de soluções gerado e devolve o peso da solução, penalizado ou não. O método *get_min_stable* verifica se o patamar de paragem é alcançado, devolvendo *True* ou *False* consoante já tenham sido, ou não, realizadas 10% do mínimo de iterações após a melhor solução atual. Os restantes métodos desta classe são de interpretação bastante direta, limitando-se a devolver os valores dos parâmetros para a iteração corrente em função das equações que dão a sua evolução.

Para realizar a mesma análise em mais do que um *core* em simultâneo foi necessário implementar uma ponte de informação entre os vários processos. A solução encontrada passa por manter a memória harmónica e a informação sobre a melhor solução em ficheiros usando a biblioteca *pickle* do Python no computador. Assim, os vários processos estão constantemente a atualizar esses dois ficheiros, partilhando as soluções entre si. É recomendado utilizar um número de iterações ligeiramente superior quando se realiza uma análise em vários *cores* para compensar eventuais perdas de informação. Não obstante, os tempos de análise poderão ser consideravelmente mais reduzidos, dependendo do número de *cores* utilizado.

4.7. OUTRAS CLASSES

Conforme referido anteriormente, o programa utiliza objetos para guardar e tratar a informação dos nós, elementos, cargas e grupos de elementos. Estas classes são mais ou menos complexas conforme as exigências que lhe são associadas. A estrutura destas será, de seguida, explicada.

4.7.1. CLASSE NODE

A classe *Node* tem como atributos o seu identificador (*tag*), as suas coordenadas (*coordx* e *coordy*), a variação de posição associada às imperfeições (*delta_x*), o seu deslocamento para as várias combinações (*displ* – uma lista com um item por combinação) e, se aplicável, a massa nodal em si concentrada (*mass*).

Esta classe tem três métodos: *coordx_os* que é utilizado na geração dos ficheiros das análises OpenSees para obter a coordenada real, *self_mass* que converte as cargas pontuais em massa e *specify_mass* que é utilizado para especificar a massa pontual, no caso de o utilizador ter especificado a massa por piso.

4.7.2. CLASSE _ELEMENT

A classe *_Element* é uma das classes mais importantes do programa, pois é responsável pela gestão de todas as verificações dentro do elemento, recorrendo ao módulo de verificações. O nome *_Element* foi

atribuído propositadamente para não ocorrerem conflitos com a classe previamente existente no módulo de verificações de segurança responsável pelas verificações à encurvadura.

Cada elemento tem como atributos o seu identificador (*tag*), os nós de extremidade (*noi* e *noj*), a sua secção (*section*), o valor do peso próprio (*PP*), o seu comprimento (*length*), uma lista com as forças internas para as várias combinações e para os vários pontos considerados (*internal_forces*) e uma lista com os deslocamentos a meio vão para as várias combinações (*mid_span_disp* – apenas se for viga).

O método ***gauss_points*** gera atualmente apenas a lista *internal_forces* em branco. Contudo, poderá ser utilizado para definir o número de pontos de Gauss caso seja pretendido utilizar elementos do tipo *forceBeamColumn* no OpenSees. O método ***inicial_sections*** atribui ao elemento a menor secção da lista de possibilidades. Este apenas tem interesse quando se utiliza o algoritmo básico. O método ***refresh_section_props*** é utilizado para atualizar, no objeto, as propriedades da secção. Sempre que esta é alterada, o método deve ser executado.

Conforme indicado no Capítulo 3, são analisadas 13 secções para as vigas e 5 secções para os pilares. Contudo, o OpenSees, uma vez que se baseia em análises de elementos finitos, apenas permite obter os valores dos esforços para as extremidades. Assim, foram implementados os métodos ***_get_internal_forces*** e ***_int_forces_aux*** que calculam os valores dos esforços para as várias secções em função das cargas aplicadas ao elemento.

Os deslocamentos a meio vão são obtidos através do método ***mid_span_displ*** que utiliza o princípio dos trabalhos virtuais. Com o objetivo de generalizar a aplicação do método a todas as configurações de diagrama, é utilizada uma regressão de 4º grau dos valores de momento nas 13 secções da viga. Naturalmente, esta regressão não permite obter o valor exato do deslocamento para todos os casos. Contudo, nas análises realizadas foram obtidos resultados muito próximos dos exatos.

À semelhança da classe *Node*, também a classe *_Element* tem um método ***self_mass*** que calcula o valor total das cargas quase permanentes da viga e obtém um valor de massa por metro, que é posteriormente utilizado pelo OpenSees na análise dinâmica.

4.7.2.1. Verificações gravíticas

A verificação dos elementos para ações gravíticas é realizada através dos métodos ***section_design_check*** e ***element_design_check***. O primeiro recebe informação sobre o material e o caso de carga e utiliza o módulo de verificações de segurança para analisar a resistência das secções do elemento para todos os pontos considerados. O segundo começa por obter as propriedades do elemento necessárias para utilizar o módulo de verificações de elementos à encurvadura. Assim, calcula o valor de C_1 através da equação (3.2) e os valores dos C_{ms} conforme indicado no Quadro B.3 do EC3, executando o módulo de seguida.

4.7.2.2. Verificações sísmicas

O método central desta verificação é ***sections_and_ele_seismic_check***, o qual altera os valores dos esforços de modo a realizar as verificações de segurança das vigas de acordo com EC8 sem ser necessário realizar alterações ao módulo de verificações. De seguida, executa as verificações seccionais e o método ***element_design_check*** que analisa os elementos à encurvadura, devolvendo um parecer e o menor valor de Ω quando utilizado em vigas e um parecer quando utilizado em pilares. Este método também chama o método ***check_duct_class_ok*** para verificar o cumprimento das condições impostas pelo quadro 6.3 do EC8, em função da classe de ductilidade pretendida.

Já, o método *ifbd_lambda* destina-se a obter o valor de λ (V_d) do elemento, no caso de utilização do IFBD no dimensionamento.

4.7.3. CLASSE LOAD

Os objetos da classe **Load** possuem como atributos a identificação do nó/elemento a que estão associados (*tag*), se é de elemento ou de nó (*tipo*), e um atributo por tipo de carga (*RCP*, *Q*, *W* e *S*). Por fim, possui um método *load_types* que devolve uma lista com os tipos de carga presentes no elemento/nó.

4.7.4. CLASSE GROUP

Os objetos da classe **Group** gerem a informação relativa aos grupos de elementos utilizados no algoritmo de pesquisa harmónica, caso sejam especificados pelo utilizador. Assim, possui como atributos a lista de elementos do grupo (*elements_list*) e a lista de secções possível para os elementos do grupo (*sec_list*). Tem ainda implementado um método que visa a alteração da secção dos elementos do grupo. Sempre que chamado este método altera a secção e atualiza as propriedades de todos os elementos do grupo.

4.8. MÓDULO DE VERIFICAÇÕES DE SEGURANÇA

Com o objetivo de manter a estrutura generalizada proposta no Capítulo 3, implementou-se um conjunto de métodos globais nas classes **Section** e **Element**. Estes métodos deverão ser suficiente para realizar qualquer análise independentemente da secção, devendo existir o cuidado de manter esta estrutura aquando da adição de novas funcionalidades. Assim, as funcionalidades base são as seguintes:

- Classe **Section**:
 - *get_properties*: deverá devolver as propriedades geométricas da secção;
 - *resistance_properties*: deverá devolver as propriedades resistentes básicas (N_{rd} , V_{rd} e M_{rd} em ambas as direções e sem interações)
 - *get_interaction_curves*: deverá representar a(s) curva(s) de interação da secção
 - *design_check*: deverá verificar a resistência da secção, em função de um conjunto de esforços e da indicação do regulamento no qual se enquadra.
- Classe **Element**:
 - *design_check*: deverá realizar a verificação de elementos uniformes em flexão composta com compressão, em função de um conjunto de esforços e de indicações do regulamento no qual se enquadra.

Com esta globalização dos métodos principais pretende-se manter a utilização do módulo simples, intuitiva e independente do tipo de secção. Naturalmente, quando forem implementadas secções de outros materiais e com exigências diferentes, poderá eventualmente ser necessária a adaptação de algum dos métodos principais ou até a adição de novos. A arquitetura global proposta para o módulo é apresentada na Fig. 4.5.

4.8.1. CLASSE SECTION

No âmbito do presente trabalho apenas foi implementada a classe *Steel_Section* e a sua ramificação *I_Commercial*. O cálculo das propriedades da secção e das curvas de interação foram desenvolvidos com base na dissertação de Barros [16], pelo que será apenas explicado o funcionamento e a interação entre os métodos.

4.8.1.1. Classe Steel_Section

A presente classe, uma sub-classe de *Section*, possui dois métodos principais: *eu_design_check*, que verifica a resistência da secção em conformidade com o EC3, e *interaction_curve_y*, que representa a curva de interação N-M na direção principal. Estes utilizam métodos da classe *Steel_Section* em conjunto com alguns métodos específicos da classe *I_Commercial* para realizar as tarefas que lhes competem. O método *eu_design_check* utiliza os métodos *ec3_design_check_N_M* e *ec3_design_check_V_M* para avaliar a resistência da secção tendo em consideração as interações de esforços axial-momento e transversal-momento, respetivamente. O primeiro tem uma morfologia muito semelhante ao método *interaction_curve_y* sendo a principal diferença o facto de devolver um parecer, *True* ou *False*, enquanto o outro devolve os pontos da curva de interação. Os restantes métodos desta classe são utilizados para obter valores resistentes da secção (*calc_nrd*, *calc_vrd* e *calc_mrd*) e na classificação das várias chapas da secção (*web_class*, *ext_comp_class* e *ext_tra_class*).

4.8.1.2. Classe I_commercial

Ao iniciar um objeto da classe *I_commercial*, uma sub-classe de *Steel_Section*, são obtidas automaticamente as principais propriedades base; as propriedades geométricas do perfil são lidas de uma base de dados, recorrendo ao método *_get_section_properties*, e a secção é classificada em função de um conjunto de esforços fornecidos, com recurso ao método *_classification_and_resistances*. Contudo, para criar uma secção, esta não precisa de estar sujeita a um conjunto de esforços. Caso o utilizador opte por não os definir a secção será classificada para regimes de compressão pura e flexão pura. Por último, são calculados os valores das resistências ao esforço axial, ao esforço transversal e ao momento plástico e elástico, para as duas direções.

O método *classification_y* é utilizado para classificar a secção em relação ao eixo principal de inércia. No caso desta ser de classe 4, é executado o método *_calc_wy_effective* para obter os valores efetivos de inércia, área e módulo de flexão efetivos. O método *_buckling_factors* é utilizado por objetos da classe *Element* para obter os fatores de imperfeição.

4.8.2. CLASSE ELEMENT

Um objeto *Element* deverá ser iniciado sendo-lhe fornecido um valor de comprimento e um objeto da classe *Section*. No âmbito do presente trabalho apenas foi implementada a classe *Steel_Element*.

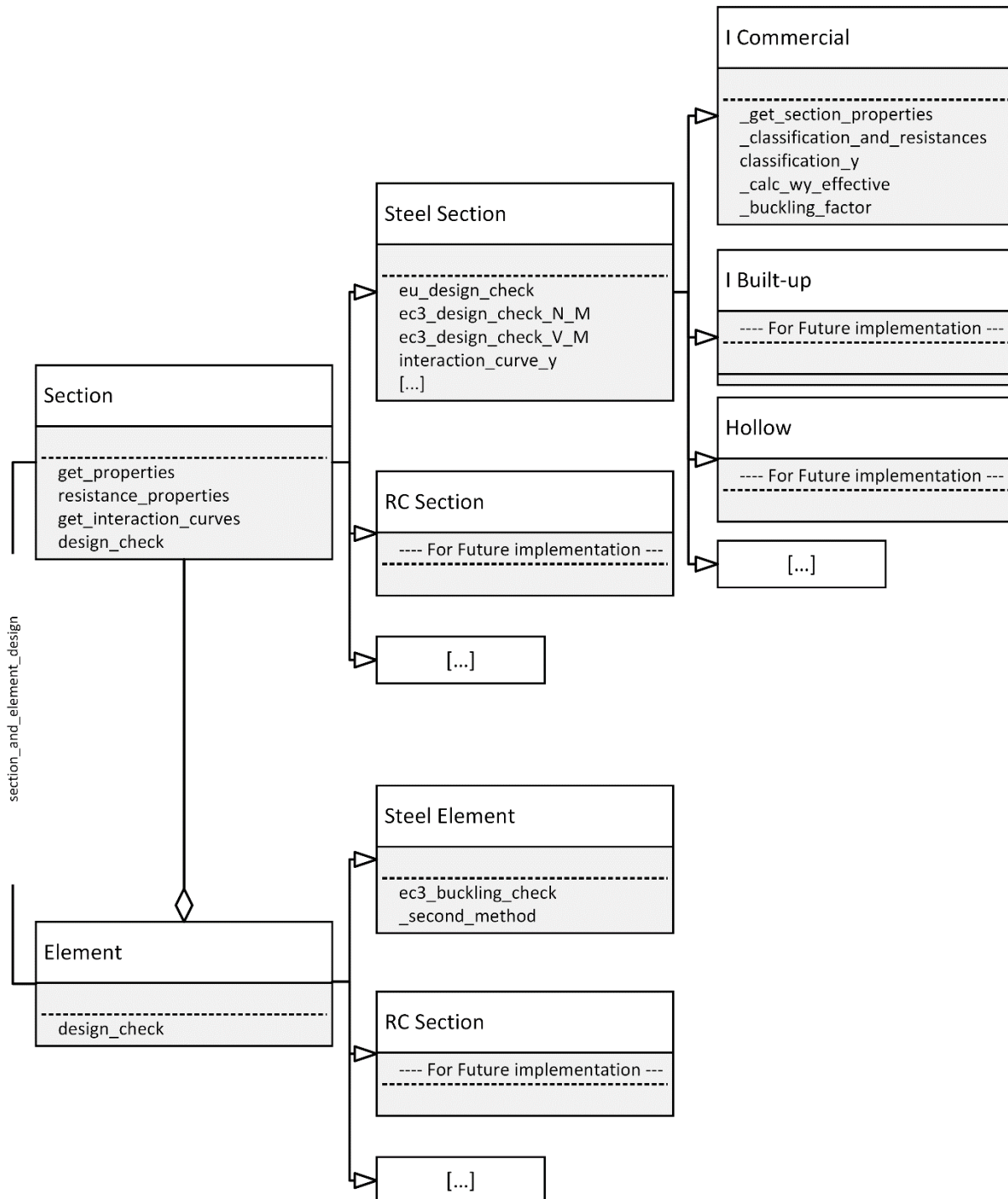


Fig. 4.5 - Diagrama de classes do módulo de verificação

4.8.2.1. Classe Steel_Element

Conforme indicado anteriormente, os objetos da classe Element apenas necessitam de verificar as condições de encurvadura no âmbito de uma regulamentação. Assim, os objetos da classe **Steel_Element** possuem um método **ec3_buckling_check** que visa averiguar a adequabilidade do elemento em relação ao bambeamento, no âmbito do EC3. Este método espera como parâmetros de introdução os valores dos coeficientes dependentes do diagrama de momento, C_1 , o valor dos fatores C_{ms} e os esforços axial e

momento. É utilizado um segundo método que visa implementar o método de cálculo dos fatores de interação, considerando que o Anexo Nacional português permite optar pelo método, apenas o Método 2 do EC3 apenas esse foi implementado. Para garantir uma fácil adaptação para problemas que utilizem o Método 1, este foi implementado num método independente, *_second_method*.

5

CASOS DE ESTUDO

5.1. INTRODUÇÃO

Neste capítulo, são apresentados os resultados da aplicação da ferramenta desenvolvida a um conjunto de pórticos metálicos simples, disponíveis em trabalhos científicos publicados recentemente. Os vários elementos dos pórticos são divididos por grupos, com o objetivo de obter soluções construtivamente sustentáveis e que otimizem, não só o peso, mas a utilização do material. Os resultados são comparados com os obtidos no Projeto Tese de Investigação de Macedo [22].

5.2. CASOS DE ESTUDO

Neste trabalho são estudados pórticos com 3 vãos e com 3, 5 e 8 pisos, dimensionados como estruturas com classe de ductilidade alta (DCH) e média (DCM), sendo as últimas dimensionadas pelo método clássico e pelo IFBD.

A estrutura do pórtico de 5 pisos é ilustrada pelo corte e pela planta apresentados na Fig. 5.1, a estrutura é regular em altura pelo que apenas varia o número de pisos nos restantes casos. O pórtico estudado está representado a tracejado na planta, sendo constituído por 3 vãos de 6 metros com cargas nos apoios e a meio vão, e por um primeiro piso de 4.5m e os restantes de 3.5m de altura.

Na Fig. 5.2 são apresentados as distribuições das ações verticais aplicadas ao pórtico e na Tabela 7 são apresentados os respetivos valores.

Relativamente às propriedades dos materiais, o aço é considerado de classe S275, tanto para vigas como pilares, com uma tensão de cedência nominal de 275 MPa. O módulo de elasticidade considerado é de 210 GPa, o módulo de distorção é de 81 GPa, o coeficiente de *Poisson* é de 0.3 e o peso volúmico de 77 kN/m³. São utilizados perfis da série IPE para as vigas e da série HEB para os pilares.

A ação sísmica é caracterizada para um solo do tipo B, considerando um amortecimento de 5% e uma classe de importância II ($\gamma_1 = 1.$). O valor de referência da aceleração máxima à superfície de um terreno do Tipo A, a_{gR} , é de 0.3 g (2,94 m/s²).

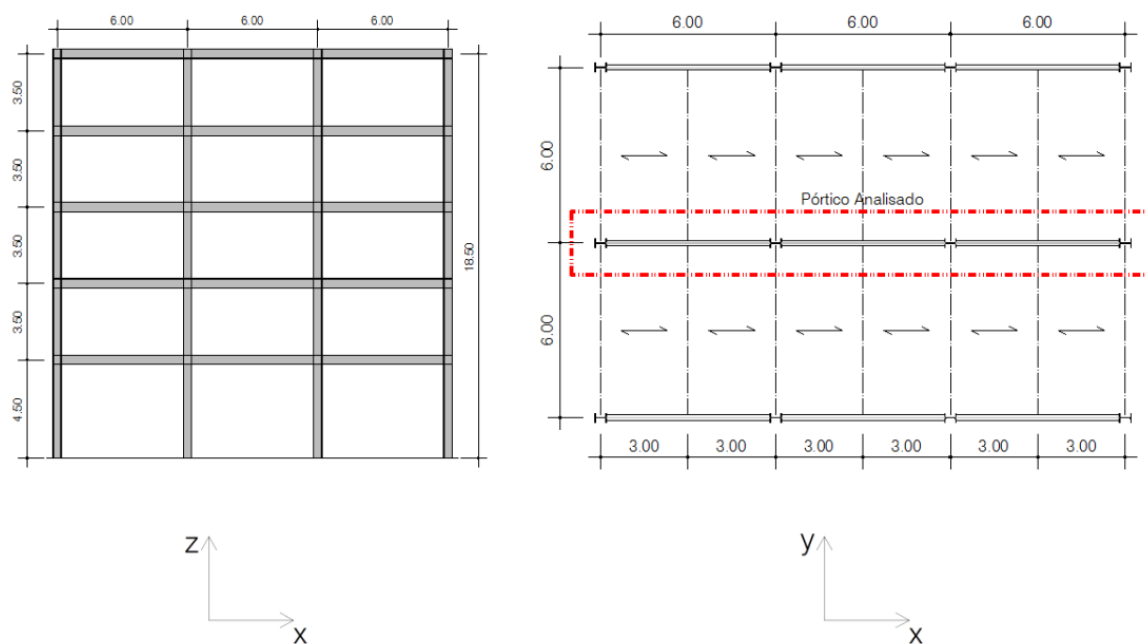


Fig. 5.1 - Representação do edifício cujo pórtico pertence, para o caso de 5 pisos. [23]

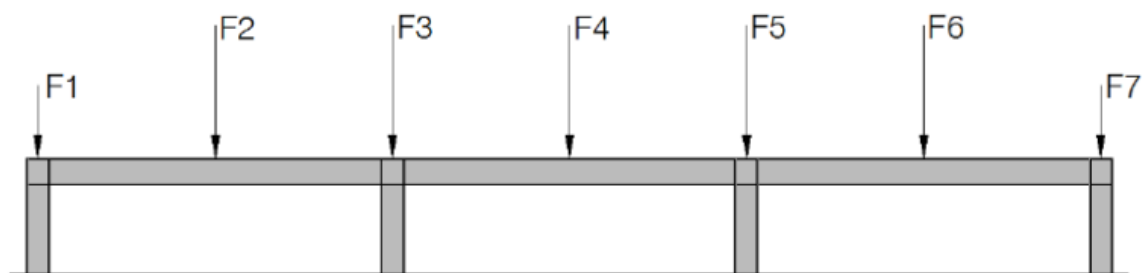


Fig. 5.2 - Distribuição das ações gravíticas num piso do pórtico [24]

Tabela 7 - Ações gravíticas para o dimensionamento do pórtico

	Carga	F1 (kN)	F2 (kN)	F3 (kN)	F4 (kN)	F5 (kN)	F6 (kN)	F7 (kN)
Cobertura	G_k	42.75	85.5	85.5	85.5	85.5	85.5	42.75
	Q_k	9	18	18	18	18	18	9
Piso Tipo	G_k	51.75	103.5	103.5	103.5	103.5	103.5	51.75
	Q_k	18	36	36	36	36	36	18

5.2.1. PÓRTICOS DE 3 PISOS

Na definição dos pórticos de 3 pisos foi adotado um agrupamento de elementos que fosse coerente com os critérios construtivos. O agrupamento de perfis considerado encontra-se representado na Fig. 5.3.

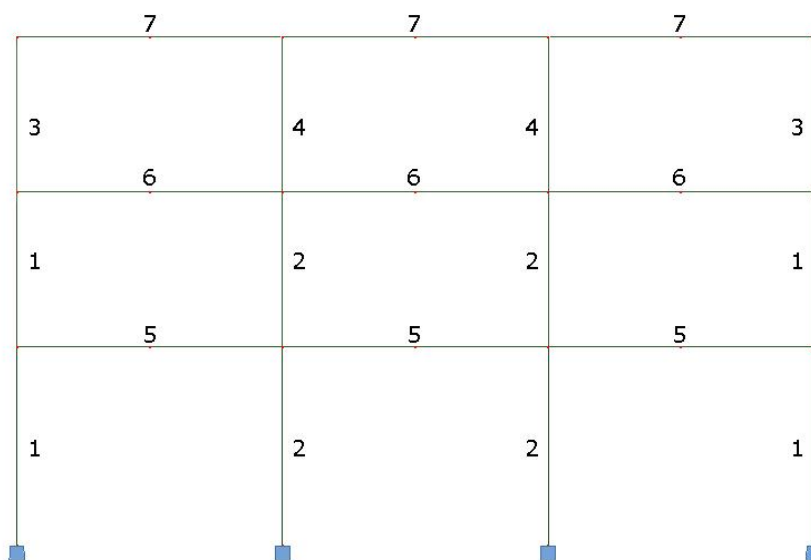


Fig. 5.3 - Definição dos grupos para o pórtico de 3 pisos

Na tabela seguinte são apresentados os resultados obtidos através de um dimensionamento manual [22] os resultados obtidos recorrendo ao programa desenvolvido neste trabalho.

Tabela 8 - Comparação dos resultados obtidos entre o dimensionamento manual e o otimizado para os pórticos de 3 pisos

	IFBD		DCM		DCH	
Grupo	Dim. Automático	Dim. Manual	Dim. Automático	Dim. Manual	Dim. Automático	Dim. Manual
1	HEB200	HEB340	HEB200	HEB360	HEB200	HEB450
2	HEB240	HEB400	HEB260	HEB400	HEB320	HEB450
3	HEB140	HEB340	HEB160	HEB360	HEB160	HEB450
4	HEB200	HEB400	HEB180	HEB400	HEB180	HEB450
5	IPE300	IPE360	IPE300	IPE360	IPE330	IPE400
6	IPE300	IPE360	IPE300	IPE360	IPE300	IPE400
7	IPE270	IPE300	IPE270	IPE300	IPE270	IPE300
Peso (ton)	5,05	9,29	5,19	9,46	5,84	10,81

Aquando da utilização do IFBD foram obtidos coeficientes de comportamento de 3.4 para o dimensionamento automático e de 1.65 para o dimensionamento manual. Contudo, a estrutura dimensionada de forma automática apresenta uma poupança de cerca de 45%, reduzindo a utilização de aço em 4.24 toneladas.

Nos dimensionamentos em que foram utilizados os valores de referência do EC8, os ganhos foram na ordem de 45% para o pórtico de ductilidade média e 46% para o pórtico de ductilidade alta. Estes ganhos correspondem a uma poupança de 4.27 e 4.97 toneladas, respetivamente.

5.2.2. PÓRTICOS DE 5 PISOS

À semelhança dos pórticos de 3 pisos, também nos de 5 foi procurado um equilíbrio entre a otimização pura e os critérios construtivos. Assim, foram definidos 11 grupos de modo a maximizar o aproveitamento dos perfis que, por norma, são fabricados com 12 metros de comprimento.

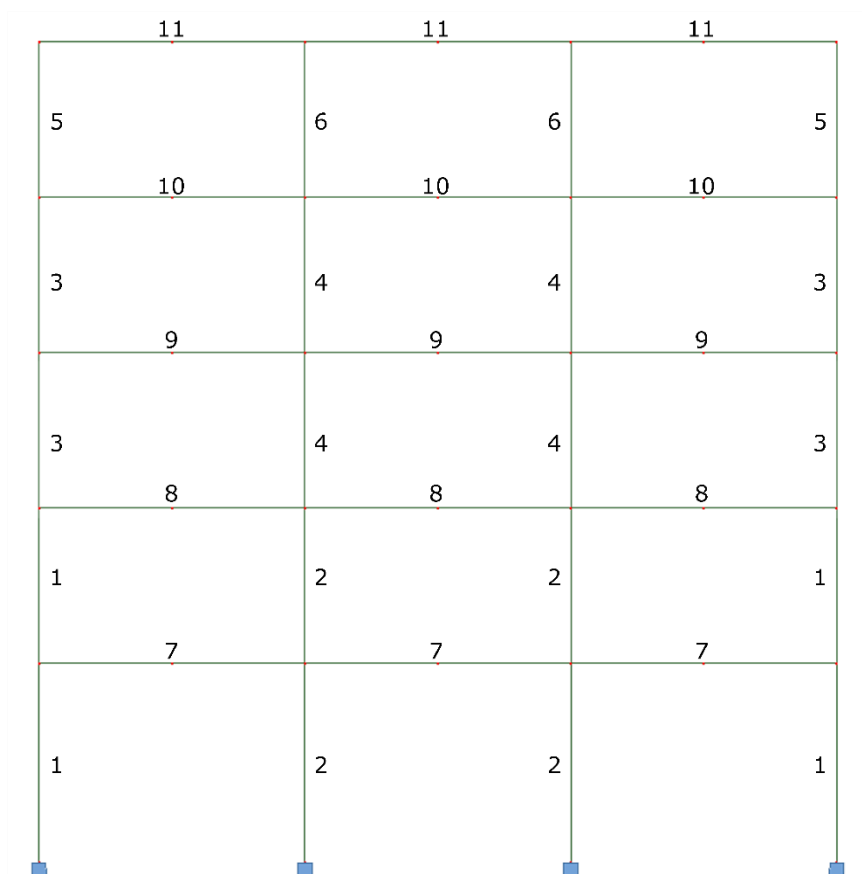


Fig. 5.4 - Definição dos grupos para o pórtico de 5 pisos

Na tabela seguinte, pode ser observada a comparação dos dimensionamentos automático e manual para os pórticos de 5 pisos. O autor dos dimensionamentos manuais [22] optou por considerar um agrupamento diferente no dimensionamento utilizando o método IFBD. Tendo considerado nos grupos 1 e 2 apenas os pilares do primeiro piso, nos grupos 3 e 4 os pilares do segundo e terceiro pisos, e nos

grupos 5 e 6 os pilares do quarto e quinto pisos. Por uma questão de normalização das soluções, neste trabalho optou-se por manter o agrupamento utilizado nos restantes pórticos de 5 pisos.

Nas estruturas dimensionadas, segundo a metodologia IFBD, foi obtido um coeficiente de comportamento de 3,4 para o dimensionamento automático e de 2.1 para o dimensionamento manual. Apesar do coeficiente de comportamento consideravelmente superior, podem ser observados ganhos de cerca de 21%, correspondendo a 2.44 toneladas por pórtico.

Tabela 9 - Comparação dos resultados obtidos entre o dimensionamento manual e o otimizado para os pórticos de 5 pisos

Grupo	IFBD		DCM		DCH	
	Dim. Automático	Dim. Manual	Dim. Automático	Dim. Manual	Dim. Automático	Dim. Manual
1	HEB200	HEB280	HEB220	HEB340	HEB280	HEB400
2	HEB300	HEB340	HEB300	HEB400	HEB360	HEB450
3	HEB200	HEB260	HEB180	HEB280	HEB240	HEB340
4	HEB240	HEB300	HEB280	HEB340	HEB280	HEB360
5	HEB160	HEB220	HEB160	HEB280	HEB160	HEB340
6	HEB200	HEB240	HEB120	HEB340	HEB180	HEB360
7	IPE330	IPE360	IPE360	IPE360	IPE400	IPE400
8	IPE330	IPE360	IPE360	IPE360	IPE400	IPE400
9	IPE300	IPE330	IPE300	IPE330	IPE360	IPE400
10	IPE300	IPE330	IPE300	IPE300	IPE300	IPE330
11	IPE270	IPE300	IPE270	IPE300	IPE270	IPE300
Peso (ton)	9,36	11,80	9,70	14,10	11,80	16,20

Os ganhos são mais expressivos para as estruturas dimensionadas utilizando os valores de referência do EC8. Assim, obteve-se ganhos na ordem dos 31% para as estruturas de ductilidade média e dos 27% para as estruturas de ductilidade alta, equivalendo a poupanças de 4.4 toneladas de aço em ambos os casos.

5.2.3. PÓRTICOS DE 8 PISOS

À semelhança dos casos anteriores, também nos pórticos de 8 pisos foi procurado um equilíbrio entre a otimização e os critérios construtivos. Assim, foram definidos 16 grupos de modo a maximizar o aproveitamento dos perfis que, por norma, são fabricados com 12 metros de comprimento. Esta divisão é apresentada na figura seguinte.

Contudo, nos dimensionamentos manuais dos pórticos de 8 pisos [22], os grupos definidos não foram transversais aos 3 pórticos considerados. Assim, embora o dimensionamento automático mantenha o

mesmo conjunto de grupos, a tabela comparativa será apresentada em função dos pisos, contrariamente aos exemplos anteriores.

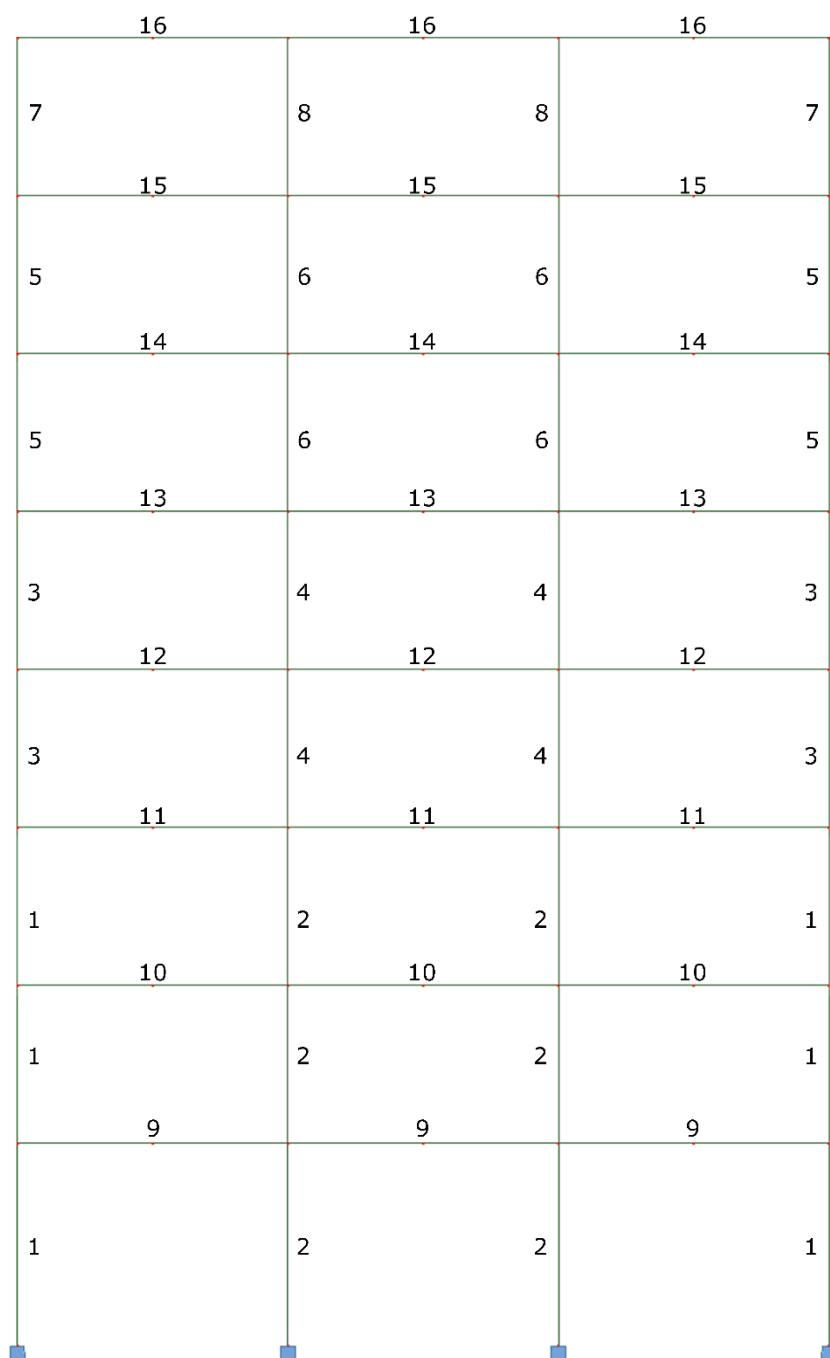


Fig. 5.5 - Definição dos grupos para o pórtico de 5 pisos

No dimensionamento utilizando a metodologia IFBD, obteve-se um ganho de cerca de 21% (4.7 toneladas). Os coeficientes de comportamento obtidos foram de 2.3 para o dimensionamento manual e 2.85 para o dimensionamento automático.

Tabela 10 - Comparação entre os dimensionamentos automático e manual utilizando a metodologia IFBD

Piso	Dimensionamento Automático			Dimensionamento Manual		
	Vigas	Pilares Internos	Pilares Externos	Vigas	Pilares Internos	Pilares Externos
1	IPE400	HEB340	HEB260	IPE400	HEB400	HEB360
2	IPE360	HEB340	HEB260	IPE360	HEB360	HEB320
3	IPE360	HEB340	HEB260	IPE360	HEB360	HEB300
4	IPE330	HEB280	HEB220	IPE360	HEB340	HEB260
5	IPE330	HEB280	HEB220	IPE330	HEB340	HEB260
6	IPE300	HEB240	HEB180	IPE330	HEB340	HEB260
7	IPE300	HEB240	HEB180	IPE330	HEB320	HEB240
8	IPE270	HEB200	HEB160	IPE300	HEB320	HEB240
Total		17,13			21,80	

De seguida, apresenta-se a tabela comparativa dos pórticos dimensionados para uma classe de ductilidade média. Constata-se ganhos de cerca de 27% na utilização do programa, o que corresponde a 7 toneladas de aço.

Tabela 11 - Comparação entre os dimensionamentos automático e manual para um pórtico de classe DCM

Piso	Dimensionamento Automático			Dimensionamento Manual		
	Vigas	Pilares Internos	Pilares Externos	Vigas	Pilares Internos	Pilares Externos
1	IPE400	HEB400	HEB300	IPE450	HEB450	HEB400
2	IPE400	HEB400	HEB300	IPE450	HEB450	HEB400
3	IPE400	HEB400	HEB300	IPE400	HEB450	HEB400
4	IPE360	HEB320	HEB220	IPE400	HEB400	HEB360
5	IPE360	HEB320	HEB220	IPE360	HEB400	HEB360
6	IPE300	HEB280	HEB180	IPE330	HEB360	HEB340
7	IPE300	HEB280	HEB180	IPE330	HEB360	HEB340
8	IPE270	HEB140	HEB160	IPE300	HEB360	HEB340
Total		19,20			26,20	

Por fim, é apresentada a tabela comparativa entre os dimensionamentos automático e manual para um pórtico de classe de ductilidade alta. Pode ser observado um ganho de cerca de 25%, correspondendo a 7.53 toneladas de aço.

Tabela 12 - Comparação entre os dimensionamentos automático e manual para um pórtico de classe DCH

Piso	Dimensionamento Automático			Dimensionamento Manual		
	Vigas	Pilares Internos	Pilares Externos	Vigas	Pilares Internos	Pilares Externos
1	IPE500	HEB450	HEB300	IPE550	HEB550	HEB500
2	IPE500	HEB450	HEB300	IPE500	HEB500	HEB450
3	IPE450	HEB450	HEB300	IPE500	HEB500	HEB450
4	IPE450	HEB360	HEB260	IPE500	HEB450	HEB400
5	IPE400	HEB360	HEB260	IPE400	HEB450	HEB400
6	IPE360	HEB300	HEB220	IPE360	HEB400	HEB360
7	IPE300	HEB300	HEB220	IPE330	HEB400	HEB360
8	IPE270	HEB200	HEB140	IPE300	HEB400	HEB360
Total		22,47			30,00	

5.3. COMPARAÇÃO DOS PÓRTICOS ESTUDADOS

Na Fig. 5.6 é apresentado um gráfico de barras que compara, em função do peso, as várias soluções obtidas segundo os métodos automático e manual. Este está organizado em função do parâmetro que motivou a definição do coeficiente de comportamento e pelo número de pisos.

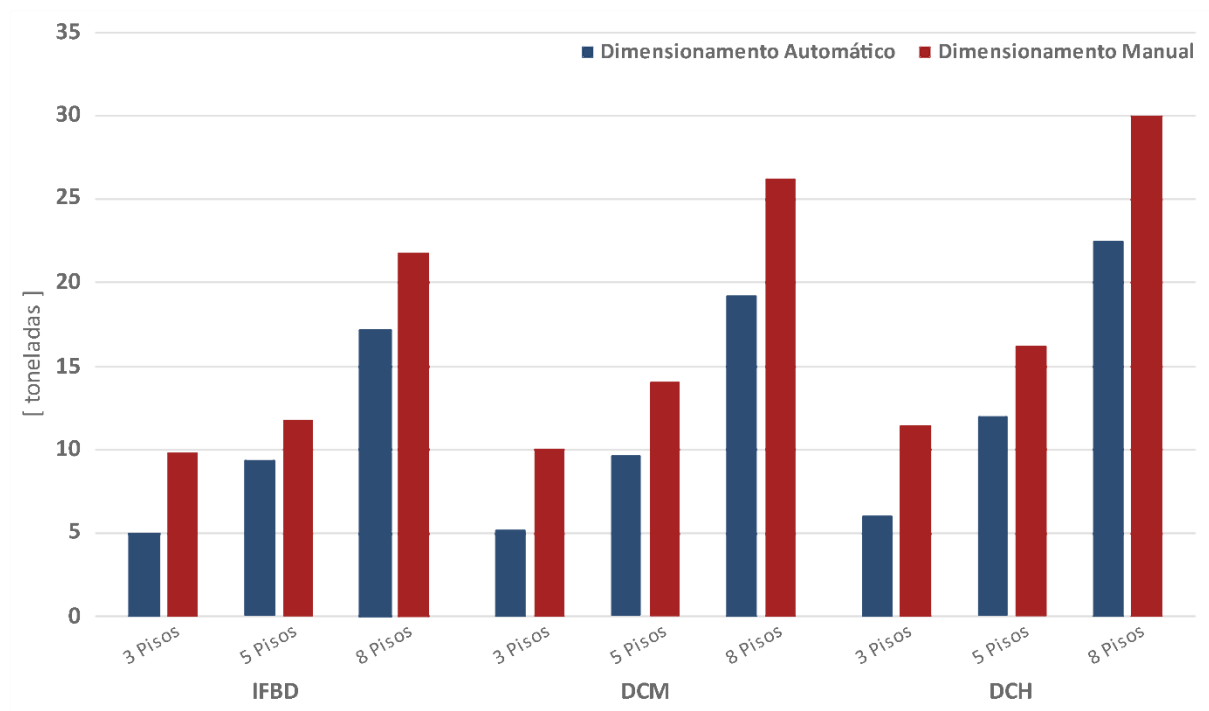


Fig. 5.6 - Gráfico comparativo dos pesos das soluções obtidas

6

CONSIDERAÇÕES FINAIS

6.1. SUMÁRIO E CONCLUSÕES

Ao longo do desenvolvimento deste trabalho foi possível criar de raiz uma ferramenta informática capaz de realizar o dimensionamento automático e otimizado de pórticos metálicos sujeitos a ações gravíticas e sísmicas. Todas as verificações de segurança previstas nos Eurocódigos 3 e 8 foram implementadas na ferramenta, incluindo a verificação de *capacity design*. De acordo com a pesquisa bibliográfica realizada, trata-se de uma ferramenta com um elevado grau de inovação, uma vez que não foi possível identificar qualquer ferramenta de dimensionamento sísmico de pórticos metálicos.

O desenvolvimento modular e *Open Source* da ferramenta concede-lhe uma grande versatilidade, permitindo a adição de novos módulos e a alteração dos parâmetros implementados de forma simples. Todas estas características associadas conferem à ferramenta grande aplicabilidade na investigação de estruturas metálicas.

O processo de otimização foi realizado com recurso à utilização de um algoritmo meta-heurístico. Nesse sentido foi realizado um estudo sobre a aplicabilidade de algoritmos de otimização a estruturas metálicas, em particular algoritmos de pesquisa harmónica. Tendo sido efetuada uma implementação e calibração dos vários parâmetros do algoritmo. Tornou-se ainda necessário o desenvolvimento de uma função objetivo baseada na proposta Camp, *et al.* [8].

Os casos de estudo apresentados no Capítulo 5 permitem concluir que a ferramenta conduz a resultados muito vantajosos quando comparados aos de um dimensionamento manual, obtendo-se poupanças de aço muito significativas. A comparação dos pórticos dimensionados recorrendo à ferramenta permitiram identificar poupanças de aço entre 20 a 50%.

Esta ferramenta é útil para o desenvolvimento de trabalhos de investigação pois permite o dimensionamento expedito de um número elevado de estruturas que possibilita a realização de estudos paramétricos à sensibilidade de parâmetros e critérios de dimensionamento.

6.2. DESENVOLVIMENTOS FUTUROS

Fruto do curto espaço de tempo atribuído à realização da tese, e apesar de ter sido possível completar o trabalho, algumas funcionalidades interessantes não puderam ser implementadas. Assim, o autor apresenta de seguida algumas propostas e intenções de desenvolvimentos futuros.

A mais importante será, talvez, a implementação de análises dinâmicas lineares com conjuntos de acelerogramas compatíveis com o espectro regulamentar. Uma vez que o método implementado (forças horizontais equivalentes) não permite uma simulação realista para edifícios altos, maiores que 8 pisos.

A implementação da ação do vento otimizada seria também de interesse para a ferramenta, uma vez que a tornaria mais completa. Esta implementação deverá ter um grau de dificuldade reduzido, pois o algoritmo encontra-se preparado para receber informações sobre as cargas resultantes da ação do vento, desde que estas sejam calculadas externamente.

Outras opções simples e interessantes passam por implementar a possibilidade de travamentos nos pilares e vigas (estando o código das últimas desenvolvido) ou implementar as condições indicadas no NA-3.2.1(1) do EC3, relativas a perfis cujos banzos têm uma espessura superior a 16mm.

Naturalmente, algumas implementações mais complexas seriam, também, interessantes, tais como a introdução de outros materiais, como estruturas de betão, ou a implementação de dimensionamento de estruturas a 3 dimensões.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Villani, A., J. Castro, and A. Elghazouli. *Improved seismic design procedure for steel moment frames*. STESSA 2009: Behaviour of Steel Structures in Seismic Areas, 2009: p. 673-678.
- [2] European Committee for Standardization *EN 1998-1, Eurocode 8: Design of Structures for Earthquake Resistance, Part 1: General Rules, Seismic Actions and Rules for Buildings*. 2004, European Committee for Standardization Brussels, Belgium.
- [3] PEER. *OpenSEES: Open System for Earthquake Engineering Simulation*. Berkeley, CA, 2006.
- [4] European Committee for Standardization *EN 1993-1-1, Eurocode 3: Design of Steel Structures. Part 1-1: General Rules and Rules for Buildings*. 2005: Comité Europeo de Normalización.
- [5] European Committee for Standardization *EN 1090-1:2009+A1:2011, Execution of steel structures and aluminium structures – Part 1: Requirements for conformity assessment of structural components*. 2011, European Committee for Standardization Brussels, Belgium.
- [6] Simões, R. *Manual de dimensionamento de estruturas metálicas, 2a edição*. Associação Portuguesa de Construção Metálica e Mista, Coimbra, Portugal, 2007.
- [7] Elghazouli, A. *Seismic design of steel structures to Eurocode 8*. Structural Engineer, 2007. **85**(12): p. 26-31.
- [8] Camp, C.V., B.J. Bichon, and S.P. Stovall. *Design of steel frames using ant colony optimization*. Journal of Structural Engineering, 2005. **131**(3): p. 369-379.
- [9] Degertekin, S. *Optimum design of steel frames using harmony search algorithm*. Structural and Multidisciplinary Optimization, 2008. **36**(4): p. 393-401.
- [10] Hasańcebi, O., F. Erdal, and M.P. Saka. *Adaptive harmony search method for structural optimization*. Journal of Structural Engineering, 2009. **136**(4): p. 419-431.
- [11] Kumar, V., J.K. Chhabra, and D. Kumar. *Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems*. Journal of Computational Science, 2014. **5**(2): p. 144-155.
- [12] Mahdavi, M., M. Fesanghary, and E. Damangir. *An improved harmony search algorithm for solving optimization problems*. Applied Mathematics and Computation, 2007. **188**(2): p. 1567-1579.
- [13] Pezeshk, S., C. Camp, and D. Chen. *Design of nonlinear framed structures using genetic optimization*. Journal of Structural Engineering, 2000. **126**(3): p. 382-388.
- [14] Saka, M.P. *Optimum design of steel sway frames to BS5950 using harmony search algorithm*. Journal of Constructional Steel Research, 2009. **65**(1): p. 36-43.
- [15] Mazzoni, S., et al. *Open System for Earthquake Engineering Simulation User Command-Language Manual*. 2009; Disponível em: <http://opensees.berkeley.edu/OpenSees/manuals/usermanual/>.
- [16] Barros, R.M.S. *Desenvolvimento de um web service para apoio ao cálculo de estruturas metálicas*. 2013.

- [17] Gardner, L., D.A. Nethercot, and H. Gulvanessian. *Designers' guide to EN 1993-1-1: eurocode 3: design of steel structures: general rules and rules for buildings*. 2005: Thomas Telford.
- [18] Geem, Z.W., J.H. Kim, and G. Loganathan. *A new heuristic optimization algorithm: harmony search*. Simulation, 2001. **76**(2): p. 60-68.
- [19] Ronacher, A. *Jinja2*. 2014; Disponível em: <http://jinja.pocoo.org/>.
- [20] Django Software Foundation. *Django*. 2015; Disponível em: <https://www.djangoproject.com/>.
- [21] Fairchild, G. *pyHarmonySearch*. 2014; Disponível em: <https://github.com/gfairchild/pyHarmonySearch>.
- [22] Macedo, L. *"Dimensionamento Sísmico de Pórticos Metálicos de Ligações Rígidas Baseado em Desempenho"* Projeto Tese de Investigação, 2015.
- [23] Arajo, M., J. Castro, and R. Delgado. *Simplified procedures for the seismic assessment of structural component demands*. in *Vienna Congress on Recent Advances in Earthquake Engineering and Structural Dynamics*. 2013.
- [24] Lopes, N.d.S. *Estudo comparativo sobre o dimensionamento sísmico de pórticos metálicos simples e contraventados*. 2013.

